



TUGAS AKHIR - TE 145561

**DESAIN PROTOTIPE DAN KOORDINASI ADAPTIF RELAI ARUS
LEBIH TERHADAP SUMBER TEGANGAN PADA SALURAN
LISTRIK SATU FASA**

Wididio Bagus Budi Arto
NRP 2214038017

Dosen Pembimbing
Dr. Eng. Ardyono Priyadi, S.T., M.Eng.

PROGRAM STUDI TEKNIK LISTRIK
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya
2017

-----Halaman ini sengaja dikosongkan-----



FINAL PROJECT - TE 145561

**PROTOTYPE DESIGN AND COORDINATIONADAPTIVE OVER
CURRENT RELAY TO VOLTAGE SOURCE ON ONE PHASE
ELECTRIC CHANNEL**

Wididio Bagus Budi Arto
NRP 2214038017

Advisor
Dr. Eng. Ardyono Priyadi, S.T., M.Eng.

ELECTRICAL ENGINEERING STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017

-----Halaman ini sengaja dikosongkan-----

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Desain Prototipe Dan Koordinasi Adaptif Relai Arus Lebih Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, **19 Juli 2017**

Mahasiswa



Wididio Bagus B. A.
NRP 2214038017

-----Halaman ini sengaja dikosongkan-----

**DESAIN PROTOTIPE DAN KOORDINASI ADAPTIF RELAI
ARUS LEBIH TERHADAP SUMBER TEGANGAN PADA
SALURAN LISTRIK SATU FASA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada**

**Program Studi Teknik Listrik
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing

Dr. Eng. Ardyono Priyadi, S.T., M.Eng.

NIP. 1973 09 27 1998 03 1004

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

SISTEM KOMUNIKASI DAN MONITORING ADAPTIF RELAI ARUS LEBIH TERHADAP SUMBER TEGANGAN PADA SALURAN LISTRIK SATU FASA

Nama : Wididio Bagus Budi Arto
NRP : 2214038017
Pembimbing : Dr. Eng. Ardyono Priyadi, S.T., MEng.
NIP : 1973 09 27 1998 03 1004

ABSTRAK

PT PLN (Persero) mengalami peningkatan jumlah pelanggan rata-rata 8,4% per tahun. Pertambahan pelanggan yang dapat diartikan sebagai pertambahan beban harus diiringi dengan pertambahan sumber agar beban pada saluran tenaga listrik dapat disuplai listrik secara maksimal dan dapat terhindar dari gangguan beban lebih pada saluran tenaga listrik.

Permasalahannya adalah, jika semakin besar beban dan sumber yang ada, maka arus nominal dan arus gangguan pada saluran juga akan semakin besar. Besarnya arus pada saluran sangat beresiko merusak peralatan listrik. Oleh karena itu dibutuhkan sebuah perangkat proteksi yang tepat seperti relai arus lebih yang nilai *setting* arusnya dapat berubah seiring dengan bertambahnya jumlah sumber atau bersifat adaptif.

Pada Tugas Akhir ini dibuat sebuah permodelan atau miniature relai arus lebih yang dapat bersifat adaptif. Relai ini dibuat sejumlah dua buah untuk mensimulasikan koordinasi proteksi pada jaringan satu fasa.

Relai tersebut dapat merubah nilai *setting* arusnya seiring dengan adanya penambahan sumber listrik. Nilai *setting* arus ini kemudian akan diatur sedemikian rupa sehingga relai dapat berkoordinasi satu sama lain dalam mengamankan jaringan satu fasa agar tidak *trip* secara bersamaan.

Kata Kunci : penambahan sumber, relai arus lebih, adaptif, koordinasi proteksi.

-----Halaman ini sengaja dikosongkan-----

ADAPTIVE OVERCURRENT RELAY COMMUNICATION AND MONITORING SYSTEM TO VOLTAGE SOURCE ON ONE PHASE ELECTRIC CHANNEL

Name : Wididio Bagus Budi Arto
NRP : 2214038017
Advisor : Dr. Eng. Ardyono Priyadi, S.T., MEng.
NIP : 1973 09 27 1998 03 1004

ABSTRACT

PT PLN (Persero) has increased the number of customers an average of 8.4% per year. Addition of subscribers which can be interpreted as the burden increase must be accompanied by the increase of source so that the load on the power line can be supplied with maximum power and can be avoided from the disruption of overload on the power line.

The problem is, if the larger the load and the available source, then the nominal current and the current interference on the channel will also be greater. The amount of current on the channel is very risky damage to electrical equipment. Therefore a precise protective device such as Over Current Relay is required that the current setting value may change as the number of sources increases or is adaptive.

In this final project, there is created a model or miniature over current relay can be adaptive. This relay is made up of two pieces to simulate the coordination of protection on a single phase network

The relay will be able to change the current setting value along with the addition of power source. The current setting values will then be arranged in such a way that relays can coordinate with each other in securing a single phase network so as not to trip simultaneously.

Keywords : *increased of source, over current relay, adaptive, ccordination protection.*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Alhamdulillah kami panjatkan kepada Allah Subhanahu Wa Ta'ala, atas limpahan rahmat dan kemudahan dariNya, hingga kami dapat menyelesaikan tugas akhir ini dengan baik, begitu pula dengan pembuatan buku tugas akhir ini.

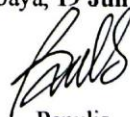
Tugas akhir ini dilakukan untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Jurusan D3 Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro dengan judul :

DESAIN PROTOTIPE DAN KOORDINASI ADAPTIF RELAI ARUS LEBIH TERHADAP SUMBER TEGANGAN PADA SALURAN LISTRIK SATU FASA

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Suwito. ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 19 Juli 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan	3
1.7 Relevansi	4
 BAB II TEORI DASAR	 5
2.1 Arduino UNO	5
2.2 Arduino MEGA 2560	6
2.3 Arduino IDE	7
2.4 Modul Komunikasi RS485	8
2.5 Relay DC 5 V	9
2.6 Relay AC M2Y 220 V	10
2.7 Real Time Clock (RTC) DS1307	10
2.8 Modul SD Card (MMC)	11
2.9 Miniature Circuit Breaker (MCB)	13
2.10 Uninterruptible Power Supply (UPS) Inforce 650 VA	13
2.11 Liquid Crystal Display Keypad (LCD Keypad)	14
2.12 Trafo (Transformator)	15
2.13 Setting Relay	16
 BAB III PERANCANGAN DAN PEMBUATAN ALAT	 19
3.1 Diagram Fungsional Alat	19
3.2 Perancangan Mekanik	21

3.3	Perancangan Elektronik.....	22
3.3.1	Perancangan Sensor Tegangan	23
3.3.2	Perancangan Sensor Arus	24
3.3.3	Perancangan <i>Relay Actuator</i>	26
3.4	Perancangan Perangkat Lunak (<i>Software</i>) Arduino IDE.....	27
3.4.1	Penrograman Sensor Tegangan	28
3.4.2	Penrograman Sensor Arus.....	30
3.4.3	Penrograman <i>Relay</i>	31
BAB IV PENGUJIAN DAN ANALISA		33
4.1	Pengujian UPS (<i>Uninterruptible Power Supply</i>)	33
4.2	Pengujian Kecepatan Respon Sistem terhadap Halangan	35
4.3	Pengujian <i>Input/Output Arduino</i>	37
4.4	Pembacaan Sensor Arus.....	42
4.5	Pembacaan Sensor Tegangan	48
4.6	Pengujian <i>Relay</i>	50
4.7	Pengujian Koordinasi Adaptif <i>Relay</i>	55
4.8	Analisa Relevansi.....	58
BAB V PENUTUP.....		59
5.1	Kesimpulan	59
5.2	Saran	59
DAFTAR PUSTAKA		61
LAMPIRAN A		A-1
LAMPIRAN B		B-1
LAMPIRAN C		C-1
DAFTAR RIWAYAT HIDUP.....		D-1

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 <i>Schematic</i> Arduino UNO	5
Gambar 2.2 <i>Schematic</i> Arduino MEGA 2560.....	6
Gambar 2.3 Jendela Arduino IDE	7
Gambar 2.4 <i>Schematic</i> Modul RS485.....	8
Gambar 2.5 <i>Schematic</i> Relay DC 5 V.....	9
Gambar 2.6 Rangkaian dalam <i>Relay</i> AC MY2 220 Volt	10
Gambar 2.7 <i>Schematic</i> RTCTiny I2C Modules	11
Gambar 2.8 <i>Schematic</i> Modul SD Card / MMC.....	12
Gambar 2.9 <i>Mini Circuit Breaker</i> (MCB).....	13
Gambar 2.10 UPS Inforce 650 VA	14
Gambar 2.11 <i>Liquid Crystal Display Keypad</i> (LCD Keypad).....	15
Gambar 2.12 Trafo <i>Step down</i> 350mA	16
Gambar 3.1 Skema Sistem Secara Keseluruhan	20
Gambar 3.2 Skema Relai.....	21
Gambar 3.3 Perancangan Mekanik Relai Utama Tampak Atas	21
Gambar 3.4 Perancangan Mekanik Relai Utama Tampak Samping	22
Gambar 3.5 Perancangan Elektronik Relai Utama	22
Gambar 3.6 Rangkaian Sensor Tegangan	23
Gambar 3.7 Rangkaian Sensor Arus	25
Gambar 3.8 <i>Schematic</i> Relay dengan Arduino.....	26
Gambar 3.9 Skema Perancangan Rangkaian <i>Relay</i>	27
Gambar 3.10 <i>Flowchart</i> Pemrograman Prototipe Relai	28
Gambar 3.11 <i>Flowchart</i> Pemrograman Sensor Tegangan	29
Gambar 3.12 <i>Flowchart</i> Pemrograman Sensor Arus SCT-013	30
Gambar 3.13 <i>Flowchart</i> Pemrograman Relai Utama	32
Gambar 4.1 Skema Pengujian UPS	34
Gambar 4.2 Bentuk Gelombang UPS sebagai Baterai.....	34
Gambar 4.3 Bentuk Gelombang UPS dengan <i>Input</i> 220 Volt	35
Gambar 4.4 Skema Pengujian <i>Power Supply</i>	35
Gambar 4.5 Bentuk Gelombang <i>Power Supply</i> Relai Utama 1	36
Gambar 4.6 Bentuk Gelombang <i>Power Supply</i> Relai Utama 2.....	36
Gambar 4.7 Hasil Pengujian <i>Power Supply</i>	37
Gambar 4.8 Skema Pengujian Pin <i>Input / Output</i> Arduino.....	37

Gambar 4.9	<i>Flowchart</i> Program Pengujian Pin Arduino (a) <i>Logic 1</i> , (b) <i>Logic 0</i>	38
Gambar 4.10	Skema Pengujian Sensor Arus	42
Gambar 4.11	Grafik Garis dan Persamaan Kalibrasi Arus Relai 1 ...	43
Gambar 4.12	Grafik Garis dan Persamaan Kalibrasi Arus Relai 2 ...	46
Gambar 4.13	Proses Pengujian Sensor Arus.....	48
Gambar 4.14	Skema Pengujian Sensor Tegangan	48
Gambar 4.15	Kurva Kerja Relai 1 Pada Sumber 1	51
Gambar 4.16	Kurva Kerja Relai 2 Pada Sumber 1	52
Gambar 4.17	Kurva Kerja Relai 1 Pada Sumber 2	53
Gambar 4.18	Kurva Kerja Relai 1 Pada Sumber 2	53
Gambar 4.19	Kurva Perbandingan Setting Relai 1	54
Gambar 4.20	Kurva Perbandingan Setting Relai 2	55
Gambar 4.21	Proses Pengujian Koordinasi Adaptif Relai	55

DAFTAR TABEL

HALAMAN

Tabel 2.1 Konfigurasi Pin LCD <i>Keypad</i> 16x2	15
Tabel 2.2 Koefisien <i>Setting Relay</i>	17
Tabel 4.1 Pengujian Tegangan Keluaran UPS	34
Tabel 4.2 Pengujian <i>Power Supply</i> Relai Utama 1	36
Tabel 4.3 Pengujian <i>Power Supply</i> Relai Utama 2	36
Tabel 4.4 Pengujian Relai Utama 1	38
Tabel 4.5 Pengujian Relai Utama 2	40
Tabel 4.6 Pengujian untuk Persamaan Kalibrasi Relai 1	42
Tabel 4.7 Pengujian untuk Data Arus Relai 1	44
Tabel 4.8 Pengujian untuk Persamaan Kalibrasi Relai 2	45
Tabel 4.9 Pengujian untuk Data Arus Relai 2	47
Tabel 4.10 Pengujian untuk Data Tegangan Relai 1	49
Tabel 4.11 Pengujian untuk Data Tegangan Relai 2	49
Tabel 4.12 Pengujian Relai 1 dengan 1 sumber	51
Tabel 4.13 Pengujian Relai 2 dengan 1 sumber	51
Tabel 4.14 Pengujian Relai 1 dengan 2 sumber	52
Tabel 4.15 Pengujian Relai 2 dengan 2 sumber	53
Tabel 4.16 Hasil Pengujian pada Koordinasi Relai 1 Sumber	56
Tabel 4.17 Hasil Pengujian pada Koordinasi Relai 2 Sumber	57

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Relai arus lebih (*Over Current Relay*) mendapatkan peran yang sangat penting pada sistem proteksi saluran tenaga listrik. Sistem proteksi ini membutuhkan keandalan yang tinggi untuk menjaga keamanan pada suatu sistem. Koordinasi waktu antar relai pada saluran merupakan hal yang sangat penting pada sistem proteksi. Selain itu, relai merupakan komponen yang bekerja berdasarkan perubahan besarnya arus.

Dalam suatu sistem yang selalu berkembang, perubahan-perubahan kondisi yang terjadi akan selalu ada. Begitu juga dengan sistem kelistrikan. Kebutuhan energi listrik pada tahun 2025 akan menjadi 457 TWh, atau tumbuh rata-rata sebesar 8,6% per tahun untuk periode tahun 2016- 2025. Sedangkan beban puncak *non coincident* pada tahun 2025 akan menjadi 74.383 MW atau tumbuh rata-rata 8,4% per tahun.[1]

Dengan pertumbuhan masyarakat saat ini yang sangat pesat, kebutuhan beban dari sistem kelistrikan pun akan terus meningkat. Namun penambahan beban ini juga harus diiringi dengan penambahan sumber. Agar beban yang semakin besar dapat tersuplai listrik. Pertambahan sumber ini menyebabkan masalah lain, yaitu pengkoordinasian setting pada alat proteksi. Jika sumber bertambah, maka setting yang ada juga harus berubah karena arus yang mengalir bertambah besar.

Perubahan besarnya arus ini disebabkan oleh bertambahnya beban yang harus disuplai oleh sumber. Sehingga jika terdapat perubahan pada sumber, otomatis *setting* relai juga berubah. Namun selama ini yang terjadi dilapangan adalah *setting* relai yang masih dilakukan secara manual, sehingga operator harus *men-setting* relai agar sesuai dengan besarnya sumber. Dan jika terdapat kesalahan *setting* saat sumber berubah, maka akan berpengaruh pada sistem pengamanan saluran.

Oleh karena itu, pada tugas akhir ini bertujuan untuk membuat sebuah permodelan yang nantinya dapat digunakan untuk mengetahui pengkoordinasian relai yang memiliki dua sumber. *Setting* relai harus dapat bersifat adaptif, yaitu dapat mengubah *setting* sesuai dengan kondisi sumber (saat sumber bertambah). Perubahan *setting* ini akan diatur oleh program yang telah dimasukkan ke *microcontroller*. Permodelan relai ini menggunakan *microcontroller* yang digunakan sebagai media *sensing*

arus dan sebagai media *setting* dan koordinasi relai untuk nantinya jika ada gangguan akan mengirim sinyal *trip* pada kontaktor.

1.2 Permasalahan

Adapun permasalahan yang akan kami angkat sebagai bahan Tugas Akhir ini :

Pertambahan beban yang ada harus diiringi dengan pertambahan sumber, agar beban yang ada dapat teraliri listrik dengan maksimal. Dengan adanya pertambahan sumber, arus yang ada pada saluran akan bertambah besar. Sehingga *setting* dari alat proteksi jaringan harus dapat bersifat adaptif. Maka dari itu, pada tugas akhir ini akan membuat permodelan *setting Over Current Relay* dengan menggunakan Arduino MEGA sebagai pemrosesan sensor arus dan *setting* yang dibutuhkan relai.

1.3 Batasan Masalah

Agar penulisan buku Tugas Akhir ini tidak menyimpang dan mengambang dari tujuan yang semula direncanakan sehingga mempermudah mendapatkan data dan informasi yang diperlukan, maka penulis menetapkan batasan-batasan masalah sebagai berikut :

- a. Perubahan *setting* relai saat penambahan sumber
- b. Gangguan berupa hubung singkat dan arus beban lebih pada jaringan satu fasa.
- c. Koordinasi adaptif relai saluran tenaga listrik satu fasa.

1.4 Tujuan

Pembuatan Permodelan *Setting Over Current Relay* yang bertujuan untuk :

1. Menentukan nilai *setting* relai untuk pengkoordinasian proteksi pada jaringan listrik satu fasa.
2. Perubahan *setting* otomatis pada *Over Current Relay* saat terjadinya penambahan sumber.
3. Relai dapat berkoordinasi antar relai dan mengamankan jaringan listrik satu fasa saat terjadi gangguan *overload* dan *shortcircuit*.

1.5 Metodologi Penelitian

Dalam pembuatan tugas akhir Prototipe *Over Current Relay* menggunakan *setting Standard Inverse*, ada beberapa tahap kegiatan yaitu meliputi tahap persiapan (*study literature*), tahap perencanaan dan pembuatan alat, tahap pengujian dan analisa, serta penyusunan laporan

Pada tahap studi literatur akan dipelajari konsep pembacaan sensor arus dan tegangan pada sumber AC, mempelajari sistematika proteksi pada tegangan listrik satu fasa, dan mempelajari mengenai konsep kerja dan koordinasi OCR(*Over Current Relay*).

Tahap perancangan *hardware* dan *software* meliputi perancangan prototype relai dilengkapi dengan sensor arus, sensor tegangan, relai dc, komunikasi RS485 dan kontaktor yang nantinya akan dikendalikan dengan mikrokontroler(Arduino Mega 2560). Pada tahap ini akan dilakukan pembuatan program pada Arduino untuk pembacaan sensor arus dan tegangan yang nantinya hasil dari pembacaan sensor tersebut akan mengendalikan relai DC untuk memicu *coil* kontaktor untuk membuka dan menutup.

Setelah itu dilakukan pengujian alat, menganalisa kesalahan atau kegagalan pada alat dan mengatasi permasalahan tersebut. Tahapan ini dilakukan dengan melakukan pengujian kerja setiap relai dan koordinasi antar relai. Data hasil pengujian tersebut akan dianalisa kemudian mencari tahu faktor apa saja yang menyebabkan alat tidak bekerja sesuai dengan keinginan atau terjadi error. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

BAB II TEORI PENUNJANG

Berisi teori penunjang yang mendukung dalam perencanaan dan pembuatan alat.

BAB III PERANCANGAN ALAT

Membahas tentang perencanaan dan pembuatan perangkat keras yang meliputi rangkaian-rangkaian, desain bangun, dan perangkat lunak yang meliputi program yang akan digunakan untuk mengaktifkan alat tersebut.

BAB IV PENGUJIAN DAN ANALISA ALAT

Membahas tentang pengukuran, pengujian, dan penganalisaan terhadap kepresisian sensor dan alat yang telah dibuat.

BAB V PENUTUP

Menjelaskan tentang kesimpulan dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.7 Relevansi

Diharapkan alat ini dapat terealisasi, alat ini dapat digunakan untuk Untuk mempermudah proses pembelajaran dalam koordinasi *setting Over Current Relay*.

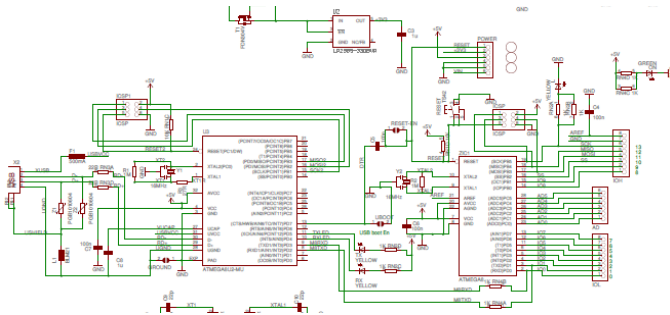
BAB II

TEORI DASAR

Pada Bab II ini akan dijelaskan mengenai teori-teori dasar yang menunjang dan berhubungan dalam pengerjaan Tugas Akhir ini. Teori dasar ini diharapkan mampu membantu dalam pengerjaan Tugas Akhir dan dapat dijadikan referensi nantinya.

2.1 Arduino UNO

Arduino UNO adalah papan mikrokontroler berdasarkan ATmega328. Arduino ini memiliki 14 pin *input / output* digital (yang 6 dapat digunakan sebagai output PWM), 6 *input* analog, resonator keramik 16 MHz, koneksi USB, colokan listrik, header ICSP, dan tombol reset. Ini berisi semua yang dibutuhkan untuk mendukung mikrokontroler. Cukup hubungkan ke komputer dengan kabel USB atau nyalakan dengan adaptor AC-ke-DC atau baterai untuk memulai.



Gambar 2.1 *Schematic* Arduino UNO

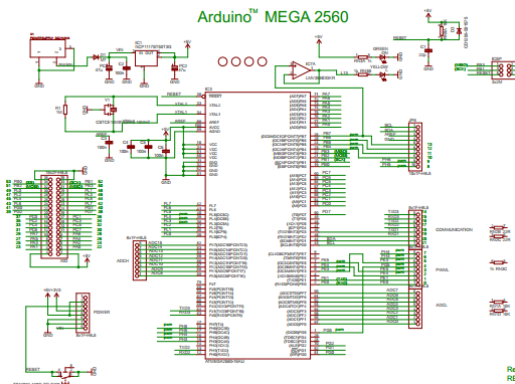
Spesifikasi Arduino Mega 2560 adalah sebagai berikut:

1. *Microcontroller* ATmega328
2. Tegangan operasi 5V
3. Tegangan input (yang direkomendasikan) 7-12V
4. Tegangan input (batas) 6-20V
5. Pin digital I/O 14 (6 dapat digunakan sebagai output PWM)
6. 6 pin analog *input*
7. Arus DC pin I/O Pin 40 mA
8. Arus DC untuk 3.3V Pin 50 mA

9. *Flash Memory* 32 KB (ATmega328) yang 0.5 KB digunakan sebagai *bootloader*
10. SRAM 2 KB (ATmega328)
11. EEPROM 1 KB (ATmega328)
12. *Clock Speed* 16 MHz

2.2 Arduino MEGA 2560

Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega2560. Arduino ini memiliki 54 digital pin input / output (yang 15 dapat digunakan sebagai output PWM), 16 analog input, 4 UART (hardware port serial), 16 MHz osilator kristal, koneksi USB, jack listrik, header ICSP, dan tombol reset. Arduino ini berisi semua yang diperlukan untuk mendukung mikrokontroler; untuk dapat terhubung ke komputer dengan menggunakan kabel USB atau sumber tegangan berasal dari adaptor AC-DC atau baterai untuk menghidupkan arduino .



Gambar 2.2 Schematic Arduino MEGA 2560

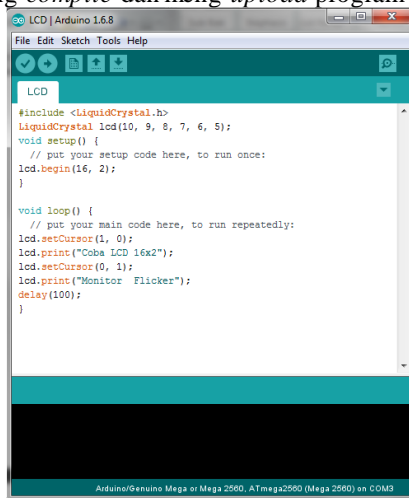
Spesifikasi Arduino Mega 2560 adalah sebagai berikut:

1. Menggunakan chip microcontroller AtMega2560.
2. Tegangan operasi 5 Volt.
3. Tegangan input (yang direkomendasikan, via jack DC) sebesar 7-12 Volt.
4. Digital I/O sebanyak 54 buah, 6 diantaranya menyediakan PWM output.
5. Analog input pin sebanyak 16 buah.

6. Arus DC per pin I/O sebesar 20 mA.
7. Arus DC pada pin 3,3 Volt sebesar 50 mA.
8. Flash memory sebesar 256 KB, 8 KB telah digunakan untuk bootloader.
9. SRAM sebesar 8 kb.
10. EEPROM sebesar 4 kb.
11. Clock speed sebesar 16 Mhz.
12. Dimensi Arduino Mega 2560 sebesar 101,5 mm x 53,4 mm.
13. Berat Arduino Mega 2560 sebesar 37 g.

2.3 Arduino IDE

Board Arduino dapat diprogram menggunakan *software open source* bawaan Arduino IDE. Arduino IDE adalah sebuah aplikasi *crossplatform* yang berbasis bahasa pemrograman *processing* dan *wiring*. Arduino IDE didesain untuk mempermudah pemrograman dengan adanya kode editor yang dilengkapi dengan *syntax highlighting*, *brace matching*, dan indentasi otomatis untuk kemudahan pembacaan program, serta dapat meng-*compile* dan meng-*upload* program ke *board* dalam satu klik.



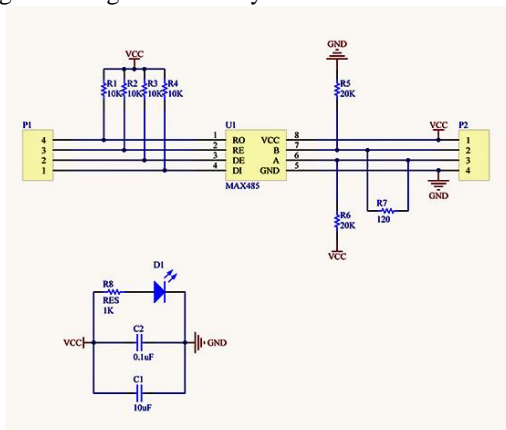
Gambar 2.3 Jendela Arduino IDE

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. *Editor program*, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah *microcontroller* tidak akan bisa memahami bahasa *processing*. Yang bisa dipahami oleh *microcontroller* adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan Arduino.

2.4 Modul Komunikasi RS485

RS485 adalah teknik komunikasi data serial yang dikembangkan di tahun 1983 dimana dengan teknik ini, komunikasi data dapat dilakukan pada jarak yang cukup jauh yaitu 1,2 Km dengan kecepatan dapat mencapai 20 Mbps. Berbeda dengan komunikasi serial RS232 yang mampu berhubungan secara *one to one*, maka komunikasi RS485 selain dapat digunakan untuk komunikasi *multidrop* yaitu berhubungan secara *one to many* dengan jarak yang jauh teknik ini juga dapat digunakan untuk menghubungkan 32 unit beban sekaligus hanya dengan menggunakan dua buah kabel saja tanpa memerlukan referensi *ground* yang sama antara unit yang satu dengan unit lainnya.



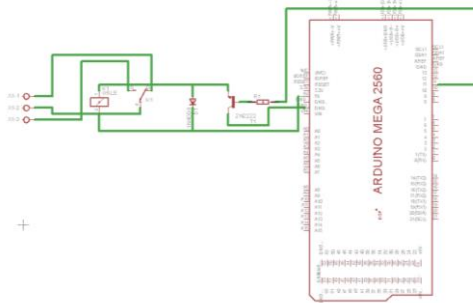
Gambar 2.4 Schematic Modul RS485

Pin yang ada pada RS485 adalah sebagai berikut :

- DI (*Data In*) Data pin DI ditransmisikan pada baris A & B saat modul berada dalam mode *Transmit* (mengirim data). Untuk mengatur modul dalam mode pengiriman nilai DE dibuat 1 dan RE dibuat 1. Pin DI terhubung ke pin Tx *Host Microcontroller* UART.
- RE (*Receive Enable*) RE pin digunakan untuk mengkonfigurasi modul dalam *Receive Mode* (menerima data).
- DE (*Data Enable*) DE pin digunakan untuk mengkonfigurasi modul dalam *Mode Transmitt*. Untuk mengfungsikan RS485 dalam mode *Transmit* dan *Receive* pin DE dan RE dihubungkan menjadi 1.
- RO (*Receive Out*) data yang diterima pada pin A & B diberikan pada pin RO. Pin RO terhubung ke pin Rx dari mikrokontroler.
- A & B (*Differential Input and Ouput Pins*) data ditransmisikan dan diterima pada garis A & B.

2.5 Relay DC 5 V

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen elektromagnetik yang terdiri dari 2 bagian utama yakni elektromagnet (*coil*) dan mekanikal (seperangkat kontak saklar/*switch*). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.



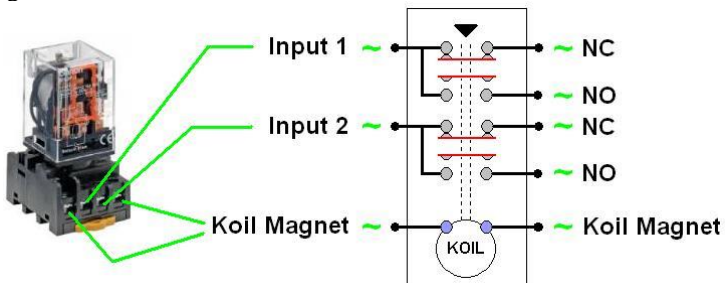
Gambar 2.5 Schematic Relay DC 5 V

AZ943 merupakan *relay* yang bekerja berdasarkan prinsip elektromagnetik, dengan tegangan operasi pada *coil* sebesar 5VDC dan arus sebesar 71,4 mA. Dengan tegangan 5 VDC, dapat membangkitkan

tegangan pada kontak sebesar 30VDC dan 277VAC. waktu operasi pada relay ini adalah 10ms (maks) dan waktu rilisnya 5ms(maks).

2.6 Relay AC M2Y 220 V

Fungsi dan cara kerja dari relay AC ini hampir sama dengan relay DC. Yang membedakan adalah sumber yang digunakan untuk mengaktifkan relay. Relay AC membutuhkan arus bolak balik, dengan tegangan antara 6 – 240 VAC.



Gambar 2.6 Rangkaian dalam Relay AC MY2 220 Volt

Relay AC MY2 memiliki spesifikasi sebagai berikut:

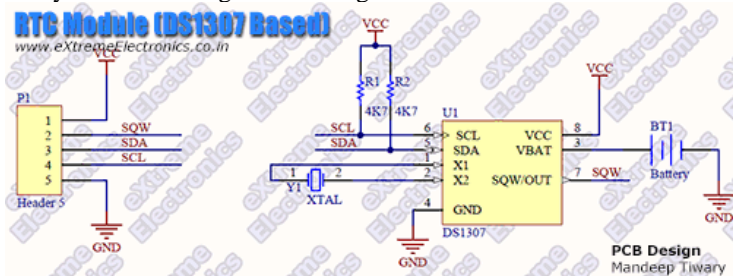
1. Tegangan kerja 220/240 VAC
2. Arus Kerja pada 50 Hz adalah 4.8/5.3 mA dan pada 60 Hz adalah 4.2/4.6 mA
3. Memiliki resistansi kumparan sebesar 18,790 Ω
4. Memiliki 8 kaki. 2 untuk *coil*, 2 untuk kontak dan memiliki 2 *Normally Open* dan 2 *Normally Close*

2.7 Real Time Clock (RTC) DS1307

Real Time Clock (RTC) adalah jenis pewaktu yang bekerja berdasarkan waktu yang sebenarnya atau dengan kata lain berdasarkan waktu yang ada pada jam kita. Meskipun istilah sering mengacu pada perangkat di komputer pribadi, server dan embedded system, RTC hadir di hampir semua perangkat elektronik yang perlu untuk menjaga keakuratan waktu. RTC memiliki sumber tenaga alternatif, sehingga mereka dapat terus menjaga waktu sementara sumber utama daya mati atau tidak tersedia. Sumber tenaga alternatif ini biasanya berupa baterai lithium dalam sistem lama, tetapi beberapa sistem yang lebih baru menggunakan supercapacitor, karena mereka dapat diisi ulang dan dapat disolder. Sumber daya alternatif juga dapat menyalurkan listrik ke RAM

yang didukung baterai. Pada umumnya tenaga alternatif yang digunakan sebesar 3 Volt dari baterai lithium.

Kebanyakan RTC menggunakan osilator kristal, tetapi beberapa menggunakan frekuensi saluran listrik. Dalam banyak kasus frekuensi osilator yang digunakan adalah 32,768 kHz. Frekuensi ini sama dengan yang digunakan dalam jam kuarsa dan jam tangan, selain itu frekuensi yang dihasilkan adalah persis 215 siklus per detik, yang merupakan tingkat nyaman untuk digunakan dengan sirkuit biner sederhana.

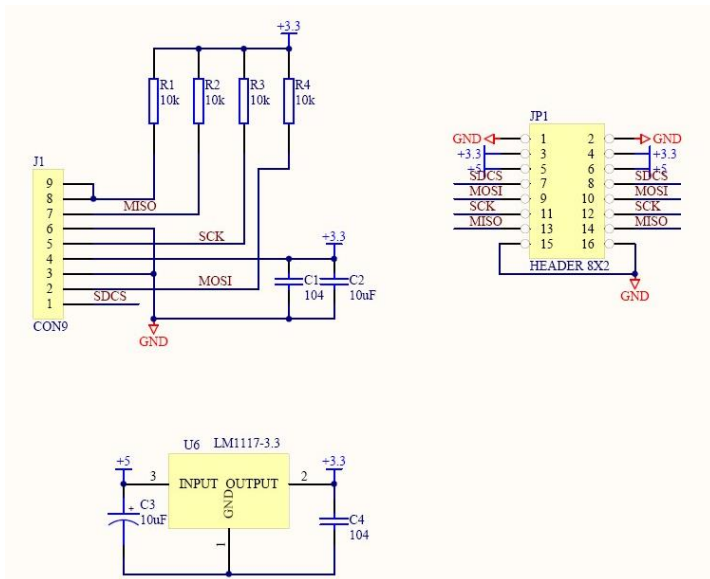


Gambar 2.7 Schematic RTC Tiny I2C Modules

Modul RTC kecil ini didasarkan pada chip jam DS1307 yang mendukung protokol I2C. RTC ini menggunakan sel baterai Lithium (CR1225). Jam / kalender menyediakan detik, menit, jam, hari, tanggal, bulan, dan Informasi tahun. Akhir tanggal bulan secara otomatis disesuaikan dengan bulan dengan kurang dari 31 hari, termasuk koreksi untuk tahun kabisat. Jam beroperasi baik dalam 24 jam atau 12 jam dengan format AM / PM indikator.

2.8 Modul SD Card (MMC)

Merupakan suatu modul untuk mempermudah antarmuka antara SD Card (atau MMC) dan mikrokontroler dengan tegangan kerja +5 VDC. SD Card (atau MMC) dapat digunakan sebagai memori yang dapat diganti dengan mudah sehingga memudahkan dalam ekspansi ke kapasitas memori yang lebih besar. Tersedia Ferroelectric Nonvolatile RAM (FRAM) yang dapat digunakan sebagai *buffer* sementara dalam mengakses SD Card (atau MMC) atau sebagai tempat penyimpanan data lain. Modul ini dapat digunakan antara lain sebagai penyimpan data pada sistem absensi, sistem antrian, atau aplikasi *data logging* lainnya.



Gambar 2.8 Schematic Modul SD Card / MMC

Modul SD Card / MMC ini memiliki spesifikasi kerja *hardware* sebagai berikut :

1. Tegangan *supply* +5 VDC.
2. Jenis kartu yang didukung: SD Card (dan MMC).
3. Antarmuka SD Card (dan MMC) dengan mikrokontroler secara SPI.
4. Tersedia 2 KByte Ferroelectric Nonvolatile RAM FM24C16.
5. Antarmuka FRAM dengan mikrokontroler secara *TwoWire Interface*.
6. Tersedia contoh aplikasi untuk DT-51™ *Low Cost Series* dan DT-AVR *Low Cost Series* dalam bahasa BASIC untuk MCS-51® (BASCOM-8051©) dan bahasa C untuk AVR® (CodeVisionAVR©).
7. Kompatibel dengan DT-51™ *Low Cost Series* dan DTA VR *Low Cost Series*. Mendukung DT-51™ *Minimum System* (MinSys) ver 3.0, DT-51™ PetraFuz, dan lain-lain.

2.9 *Miniature Circuit Breaker (MCB)*

Mini Circuit Breaker (MCB) memiliki fungsi sebagai alat pengaman arus lebih. MCB ini memproteksi arus lebih yang disebabkan terjadinya beban lebih dan arus lebih karena adanya hubungan pendek. Prinsip dasar kerjanya yaitu untuk pemutusan hubungan yang disebabkan beban lebih dengan relai arus lebih sesaat menggunakan elektromagnet.

Bila elektromagnet bekerja, maka akan memutus hubungan kontak yang terletak pada pemadam busur dan membuka saklar. MCB untuk rumah seperti pada pengaman lebur diutamakan untuk proteksi hubungan pendek, sehingga pemakaiannya lebih diutamakan untuk mengamankan instalasi atau konduktornya. Arus nominal yang digunakan pada APP dengan mengenal tegangan 230/400 V ialah: 1 A, 2 A, 4 A, 6 A, 10 A, 16 A, 20 A, 25 A, 35 A, dan 50 A disesuaikan dengan tingkat VA konsumen. Adapun kemampuan membuka (*breaking capacity*) bila terjadi hubung singkat 3 KA dan 6 KA (SPLN 108-1993). MCB yang khusus digunakan oleh PLN mempunyai tombol biru. MCB pada saat sekarang ini paling banyak digunakan untuk instalasi rumah, instalasi industri maupun instalasi gedung bertingkat.



Gambar 2.9 *Mini Circuit Breaker (MCB)*

2.10 *Uninterruptible Power Supply (UPS) Inforce 650 VA*

UPS adalah singkatan dari *uninterruptible power supply* sebagai alat *back-up* listrik ketika PC atau kehilangan energi dari sumber utamanya. UPS bekerja diantara *device* yang di suplai dan colokan listrik, dari colokan listrik yang di alirkan ke baterai yang berada pada UPS dan kemudian di simpan untuk kestabilan tegangan energi. Listrik yang di

simpan pada baterai akan di pakai ketika sumber energi utama listrik terputus.



Gambar 2.10 UPS Inforce 650 VA

Uninterruptible Power Supply (UPS) Inforce 650 VA memiliki spesifikasi sebagai berikut :

1. Kapasitas daya 650 VA
2. *Range* input 140V – 300VAC
3. Stabilitas *output* 230 V + /- 10%
4. 3 Step AVR
5. Efisiensi tinggi pada mode *in-line* 95%
6. Tipe baterai 8.2 Ah
7. 2 soket output
8. Indikator LCD

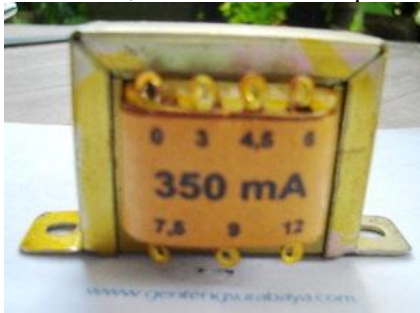
2.11 Liquid Crystal Display Keypad (LCD Keypad)

LCD (*Liquid Crystal Display*) berfungsi menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi microcontroller. Sumber cahaya di dalam sebuah perangkat LCD adalah lampu neon berwarna putih dibagian belakang susunan kristal cair tadi.

Titik cahaya yang jumlahnya puluhan ribu bahkan jutaan inilah yang membentuk tampilan. Kutub kristal cair yang dilewati arus listrik akan berubah karena pengaruh polarisasi medan magnetik yang timbul dan oleh karenanya akan hanya beberapa warna.

LCD membutuhkan driver supaya bisa dikoneksikan dengan sistem minimum dalam suatu microcontroller. Driver yang disebutkan berisi rangkaian pengaman, pengatur tingkat kecerahan maupun data, serta untuk mempermudah pemasangan di microcontroller.

elektromagnetik. Tegangan masukan bolak-balik yang membentangi primer menimbulkan fluks magnet yang idealnya semua bersambung dengan lilitan sekunder. Fluks bolak-balik ini menginduksikan gaya gerak listrik (ggl) dalam lilitan sekunder. Jika efisiensi sempurna, semua daya pada lilitan primer akan dilimpahkan ke lilitan sekunder. Transformator step-down memiliki lilitan sekunder lebih sedikit daripada lilitan primer, sehingga berfungsi sebagai penurun tegangan. Transformator jenis ini sangat mudah ditemui, terutama dalam adaptor AC-DC.



Gambar 2.12 Trafo Step down 350mA

2.13 Setting Relay

Aplikasi perhitungan dahulu digunakan untuk menentukan settingan yang sesuai pada rele konvensional, namun untuk rele saat ini sudah banyak yang memakai rele digital dimana settingan dapat diatur dan dikontrol dengan *microcontroller*. Tetapi perhitungan settingan rele OCR tetap dilakukan sebagai ilmu dasar untuk melakukan teknik men-setting suatu relai. Berikut merupakan rumus waktu kerja relai yang dipakai dan tabelnya :

$$t = \frac{\alpha \times tms}{\left\{ \left(\frac{If}{Iset} \right)^\beta - 1 \right\}}$$

Keterangan :

If = Arus gangguan

Iset = Arus setting

tms = time dial

α = koefisien kali

β = koefisien pangkat

$$tms = \frac{t x \{(\frac{If}{Iset})^\beta - 1\}}{\alpha}$$

Tabel 2.2 Koefisien *Setting Relay*

KARAKTERISTIK RELAY	α	β
STANDARD INVERSE	0.14	0.02
LONG TIME INVERSE	120	1
EXTREMELY INVERSE	80	2
VERY INVERSE	13.5	1
SHORT TIME INVERSE	0.05	0.04

-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

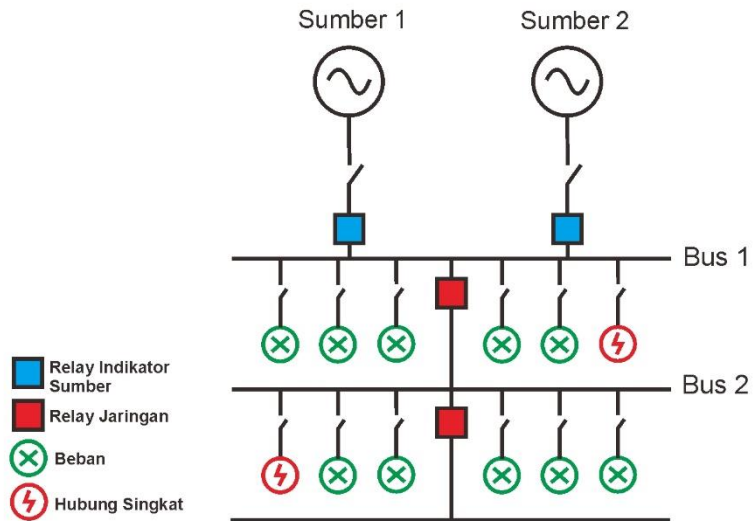
Pada bab ini akan dibahas mengenai perancangan serta pembuatan **“Rancang Bangun Koordinasi Adaptif Relai Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa”**, baik perancangan perangkat elektronik (*hardware*), perancangan dan pembuatan perangkat lunak (*software*) yang meliputi :

1. Perancangan Mekanik
2. Perancangan *Hardware* terdiri dari :
 - a Perancangan Sensor Tegangan
 - b Perancangan Sensor Arus
 - c Perancangan Relay dan MCB
 - d Perancangan Shield Arduino Mega
3. Perancangan *Software* yang berupa *flowchart* Pemrograman Arduino IDE

3.1 Diagram Fungsional Alat

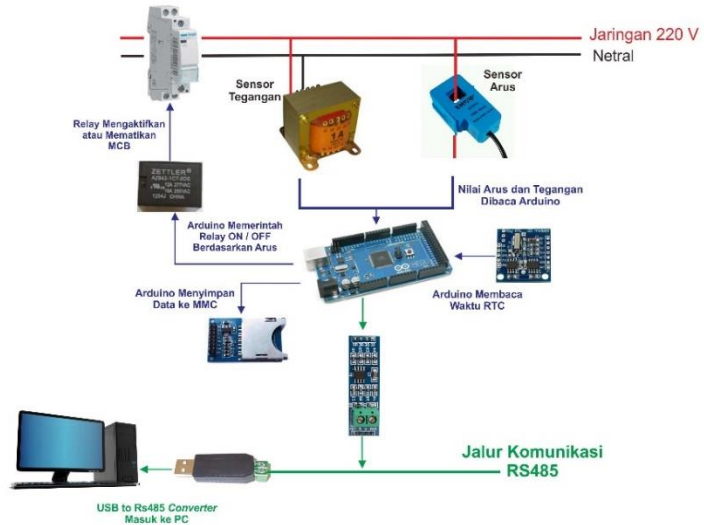
Perencanaan Tugas Akhir **“Rancang Bangun Koordinasi Adaptif Relai Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa”** ini mengenai sistem kerja alat secara keseluruhan. Dimana pada alat ini berfungsi sebagai monitor keadaan arus dan tegangan jaringan dan mengamankan dari adanya arus lebih yang disebabkan hubung singkat. Koordinasi relai dalam sistem memiliki arti bahwa dalam pemasangan relai harus memiliki koordinasi waktu agar nantinya dapat mengamankan jaringan yang memiliki gangguan dan tidak mengganggu jaringan lainnya. Sedangkan adaptif berarti bahwa *setting* waktu relai ini dapat berubah kondisi sesuai dengan keadaan sumber dan arus, jika beban dan sumber pada sistem bertambah maka besarnya arus dalam jaringan juga bertambah, agar sistem tetap berjalan dan beban tetap dapat beroperasi maka *setting* dari relai harus berubah.

Untuk dapat mengetahui apakah sumber yang digunakan hanya 1 sumber atau 2 sumber, dideteksi dengan relai indikator. Relai indikator ini berfungsi untuk memeriksa apakah sumber yang tersambung pada jaringan hanya 1 sumber atau 2 sumber.



Gambar 3.1 Skema Sistem Secara Keseluruhan

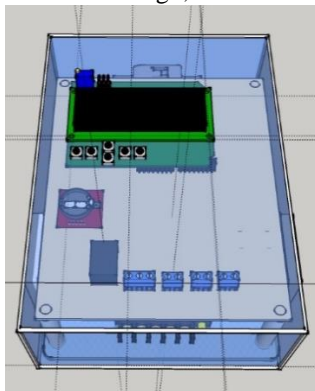
Sistem kerja dari relai ini dimulai dengan mengambil nilai dari sensor arus dan sensor tegangan, setelah terdeteksi bahwa arus pada jaringan adalah arus nominal, maka MCB akan *close* dan menyambungkan jaringan. Selanjutnya RTC akan membaca waktu dan tanggal secara *real time*, ini digunakan untuk pendataan kondisi dari jaringan agar dapat dipantau *user* secara *real time*. LCD pada relai akan menampilkan tegangan dan arus saat ini yang terukur dari jaringan. Kondisi tersebut akan disimpan di data *logger* dan akan dikirimkan ke PC untuk memonitor secara langsung. Pada PC ini dengan menggunakan software LabVIEW akan menampilkan grafik arus terhadap waktu dan tegangan yang ada pada relai.



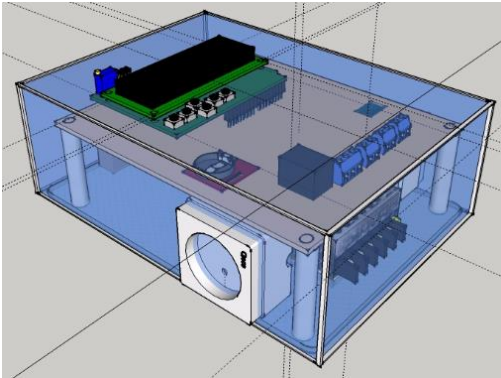
Gambar 3.2 Skema Relai

3.2 Perancangan Mekanik

Pada perancangan mekanik bagian relai utama merupakan kotak yang isinya terdiri dari komponen Rangkaian Sensor Tegangan, Rangkaian Sensor Arus, LCD Keypad 16 x 2, Komunikasi RS485, Relay DC 5 Volt, MCB(Kontaktor), Rangkaian RTC, Rangkaian SD Card, Shield Arduino Mega, dan Power Supply 12 Volt.



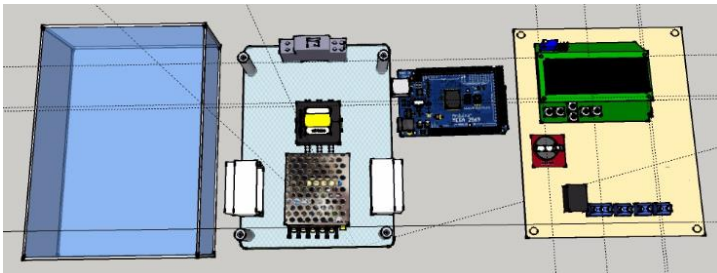
Gambar 3.3 Perancangan Mekanik Relai Utama Tampak Atas



Gambar 3.4 Perancangan Mekanik Relai Utama Tampak Samping

3.3 Perancangan Elektronik

Pada perancangan elektronik (*hardware*) pada Relai utama ada 3 bagian, yaitu perancangan sensor tegangan, perancangan sensor arus, dan perancangan *relay actuator*. Ketiga bagian tersebut merupakan bagian inti dari Relai utama. Sensor arus dan tegangan akan membaca kondisi arus dan tegangan secara *real time* dan mengirimkan sinyal ke *relay actuator* sehingga *relay actuator* membantu membuka dan menutup MCB(kontaktor) yang nantinya akan menyambung dan memutuskan saluran listrik untuk memproteksi beban yang ada pada saluran tersebut. Tampilan perancangan *hardware* untuk relai utama dapat dilihat pada gambar dibawah ini :



Gambar 3.5 Perancangan Elektronik Relai Utama

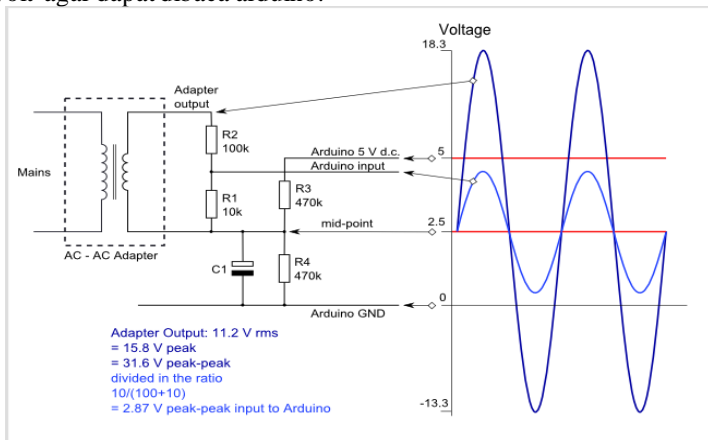
Susunan dari perancangan elektronik terbagi menjadi 2 bagian, yaitu, bagian atas dan bagian bawah. Pada bagian bawah terdapat *Power Supply*, CT Sensor, Trafo sebagai sensor tegangan, MCB dan *Stop Contact* untuk

menghubungkan jaringan dengan MCB yang ada pada relai. Sedangkan pada bagian atas terdapat 2 sisi bagian, pada sisi bawah terdapat *board* Arduino sebagai kontrol pada Tugas Akhir ini, sedangkan pada sisi atas terdapat rangkaian Sensor Tegangan, rangkaian Sensor Arus, rangkaian Relai penggerak MCB, LCD *Keypad* 16 x 2, Komunikasi RS485, RTC dan *SD Card*.

3.3.1 Perancangan Sensor Tegangan

Tugas akhir ini menggunakan sensor tegangan dari *transformer step down* 220 Volt / 12 Volt. Trafo *stepdown* digunakan untuk menurunkan tegangan 220 Volt agar nantinya dapat dibaca oleh ADC pada arduino yang membutuhkan tegangan 0-5 Volt.

Agar hasil dari sensor tegangan dapat dibaca oleh ADC arduino maka dilakukan pembagian tegangan agar didapat nilai tegangan yang lebih kecil. Tegangan yang akan masuk pada ADC masih berupa gelombang sinus yang memiliki nilai negatif. Agar semua data dapat dibaca oleh arduino, nilai negatif pada tegangan harus diubah mejadi positif. Berikut adalah gambar rangkaian untuk mengkonversi tegangan 12 Volt agar dapat dibaca arduino:



Gambar 3.6 Rangkaian Sensor Tegangan

Berdasarkan referensi dari web <https://openenergymonitor.org/>, agar dapat dibaca ADC arduino, tegangan 12 Volt dari output trafo ditransformasikan dengan cara pembagian tegangan seperti berikut:

$$\mathbf{V_{out} = \frac{R1}{R1+R2} \times V_{in} \dots\dots\dots (3.1)}$$

$$\mathbf{V_{out} = \frac{10K}{10K+100K} \times 12 \text{ Volt}}$$

$$\mathbf{V_{out} = 1,09 \text{ Volt}}$$

Untuk mengubah tegangan negatif menjadi tegangan positif dengan menggunakan rangkaian clamper yang akan menggeser sinyal. Pertama dengan membuat tegangan *mid-point* dengan menggunakan 2 resistor yang sama besar :

$$\mathbf{V_{mid} = \frac{R1}{R1+R2} \times V_{in \text{ mid}} \dots\dots\dots (3.2)}$$

$$\mathbf{V_{mid} = \frac{470k}{470k+470k} \times 5 \text{ Volt}}$$

$$\mathbf{V_{mid} = 2,5 \text{ Volt}}$$

Dengan begitu dapat disimpulkan bahwa tegangan yang masuk ke ADC Arduino Mega 2560 adalah sebagai berikut:

$$\mathbf{V_{pick \text{ to pick}} = (2 \times V_{out}) + V_{mid} \dots\dots\dots (3.3)}$$

$$\mathbf{V_{pick \text{ to pick}} = (2 \times 1,09) + 2,5 \text{ Volt}}$$

$$\mathbf{V_{pick \text{ to pick}} = 4,68 \text{ Volt}}$$

$$\mathbf{V_{pmax} = V_{mid} + V_{out} \dots\dots\dots (3.4)}$$

$$\mathbf{V_{pmax} = 2,5 \text{ Volt} + 1,09 \text{ Volt}}$$

$$\mathbf{V_{pmax} = 3,59 \text{ Volt}}$$

$$\mathbf{V_{pmin} = V_{mid} - V_{out} \dots\dots\dots (3.5)}$$

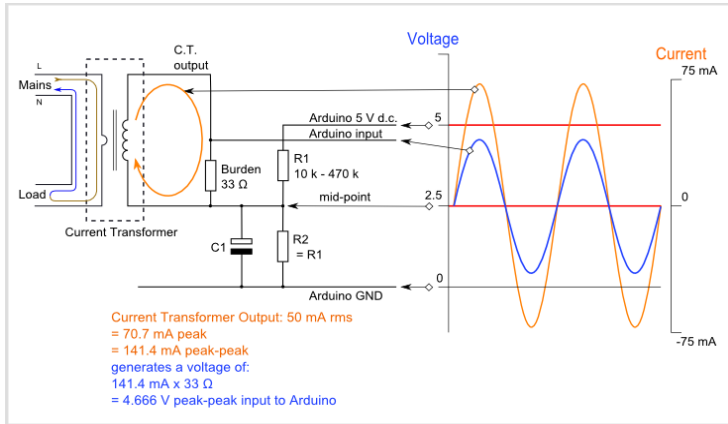
$$\mathbf{V_{pmin} = 2,5 \text{ Volt} - 1,09 \text{ Volt}}$$

$$\mathbf{V_{pmin} = 1,41 \text{ Volt}}$$

3.3.2 Perancangan Sensor Arus

Sensor arus yang digunakan pada Tugas Akhir ini adalah sensor SCT013-10. Sensor ini dapat membaca arus hingga 10 Ampere dan akan mengubahnya menjadi tegangan sebesar 1 Volt. Sensor SCT013-10 dapat mengubah output menjadi tegangan karena dalam rangkaian sensor sudah terdapat resistor *burden* sebesar 180 Ω. Sehingga arus sebesar 10 A saat melewati *R burden* akan berubah menjadi tegangan 1 Volt.

Tegangan 1 Volt dari output sensor masih memiliki tegangan negatif. Tegangan ini yang nantinya akan dimasukkan pada ADC arduino. Agar semua data dapat dibaca oleh arduino, nilai negatif pada tegangan harus diubah menjadi positif. Berikut adalah gambar rangkaian untuk mengkonversi tegangan agar dapat dibaca arduino:



Gambar 3.7 Rangkaian Sensor Arus

Berdasarkan referensi dari web <https://openenergymonitor.org/>, dengan sensor SCT13-10 yang memiliki Rasio Transformasi sebesar 1800 dan $R_{burden} = 180\Omega$, dapat diketahui besarnya arus yang ditransformasikan sensor sebelum menjadi tegangan adalah sebagai berikut :

$$I_{output\ CT} = \frac{I_{input}}{rasio} \dots\dots\dots(3.6)$$

$$I_{output\ CT} = \frac{10\ A}{1800}$$

$$I_{output\ CT} = 0,0055\ A$$

Tegangan yang dihasilkan oleh sensor arus setelah melewati resistor *burden* adalah sebesar:

$$V_{CT} = R_{burden} \times Arus\ CT \dots\dots\dots(3.7)$$

$$V_{CT} = 180\Omega \times 0,0055\ A$$

$$V_{CT} = 0,99\ Volt\ (1\ Volt)$$

Untuk mengubah tegangan negatif menjadi tegangan positif dengan menggunakan rangkaian clamper yang akan menggeser sinyal. Pertama dengan membuat tegangan *mid-point* dengan menggunakan 2 resistor yang sama besar :

$$V_{mid} = \frac{R1}{R1 + R2} \times V_{in\ mid} \dots\dots\dots(3.8)$$

$$V_{mid} = \frac{10\ K}{10\ K + 10\ K} \times 5\ Volt$$

$$V_{mid} = 2,5\ Volt$$

Dengan begitu dapat disimpulkan bahwa tegangan yang masuk ke ADC Arduino Mega 2560 adalah sebagai berikut:

$$V_{pick\ to\ pick} = (2 \times V_{out}) + V_{mid} \dots\dots\dots (3.9)$$

$$V_{pick\ to\ pick} = (2 \times 1) + 2,5 \text{ Volt}$$

$$V_{pick\ to\ pick} = 4,5 \text{ Volt}$$

$$V_{pmax} = V_{mid} + V_{out} \dots\dots\dots (3.10)$$

$$V_{pmax} = 2,5 \text{ Volt} + 1 \text{ Volt}$$

$$V_{pmax} = 3,5 \text{ Volt}$$

$$V_{pmin} = V_{mid} - V_{out} \dots\dots\dots (3.11)$$

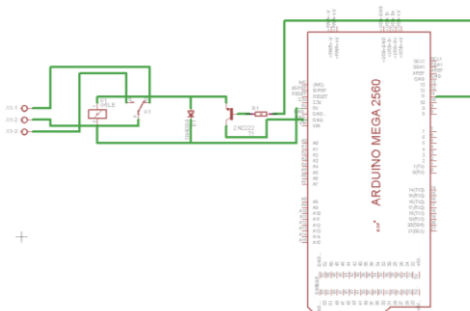
$$V_{pmin} = 2,5 \text{ Volt} - 1 \text{ Volt}$$

$$V_{pmin} = 1,5 \text{ Volt.}$$

3.3.3 Perancangan *Relay Actuator*

Relay actuator pada Tugas Akhir ini berguna untuk mengirimkan sinyal pada pemutus jaringan saat terdapat gangguan yang terbaca oleh arduino. Saat arduino membaca arus gangguan dari sensor CT, arduino akan mengaktifkan *relay actuator* yang nantinya akan memerintah MCB untuk memutuskan saluran.

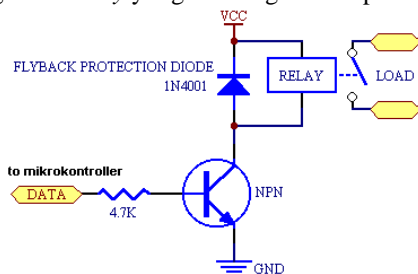
Relay DC 5 Volt akan dihubungkan pada pin ADC Arduino. Namun *relay* ini tidak mampu bekerja hanya dengan mengandalkan sinyal dari pin ADC, sehingga dipasanglah transistor sebagai *driver* relay. *Driver* ini berfungsi untuk menggerakkan *coil* yang ada pada *relay* DC 5 Volt (*actuator*).



Gambar 3.8 *Schematic Relay* dengan Arduino

Transistor yang digunakan adalah transistor NPN. Basis dari transistor dihubungkan dengan resistor dan masuk ke ADC dari Arduino.

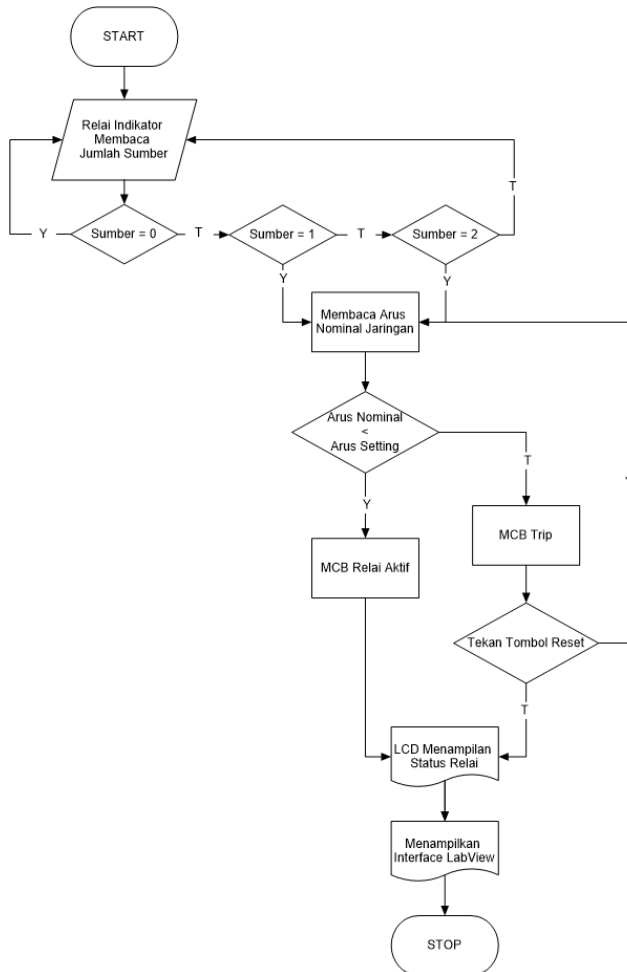
Pin kolektor terhubung dengan dioda dan terhubung pada *coil relay*. Dioda disini berguna untuk mencegah arus balik terpengaruhnya arduino dari lonjakan arus yang terjadi ketika *coil relay* bekerja. Sedangkan pin emitter terhubung dengan ground arduino. Berikut adalah gambar rangkaian *relay* yang akan digunakan pada tugas akhir ini:



Gambar 3.9 Skema Perancangan Rangkaian *Relay*

3.4 Perancangan Perangkat Lunak (*Software*) Arduino IDE

Software Arduino IDE pada Tugas Akhir digunakan untuk melakukan pemrograman papan Arduino dalam menjalankan sistem secara keseluruhan. *Software* ini menggunakan bahasa pemrograman C. Perancangan dari pemrograman ini seperti yang ditunjukkan pada Gambar 3.10, dimana ketika program dijalankan relai indikator akan membaca banyaknya sumber. Jika sumber tidak ada maka relai indikator terus membaca hingga sumber terdeteksi. Jika sumber yang melewati jaringan telah terdeteksi maka relai indikator akan mengirimkan kondisi sumber ke relai utama. Selanjutnya relai utama akan membaca data dari relai indikator dan membaca besar arus yang terdapat dalam jaringan. Dari data sumber yang didapat dari relai indikator, relai utama dapat menentukan arus *setting*-nya. Jika arus nominal jaringan kurang dari arus *setting*, maka MCB relai utama akan terhubung. Namun jika arus nominal lebih besar dari arus *setting* maka relai utama akan mendeteksi gangguan arus lebih dan MCB akan *trip* atau memutuskan jaringan. Jika tombol reset relai ditekan maka relai akan membaca arus pada jaringan lagi. Data yang ada pada relai akan ditampilkan pada LCD, disimpan pada data logger dan dikirimkan ke labVIEW sebagai *human interface*. Untuk lebih ringkasnya dapat dilihat pada *flowchart* pemrograman Arduino IDE dibawah ini :

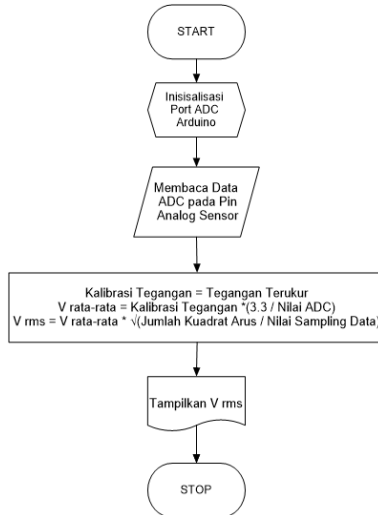


Gambar 3.10 Flowchart Pemrograman Prototipe Relai

3.4.1 Pemrograman Sensor Tegangan

Sensor tegangan yang digunakan pada Tugas Akhir ini menggunakan trafo CT 350 mA dengan tegangan 230 Volt / 9 Volt. Dimana *output* dari trafo ini berupa tegangan 9 Volt yang memiliki siklus negatif. Dengan konversi yang telah dijelaskan pada perancangan

hardware tepatnya pada persamaan 3.1, 3.2 dan 3.3 maka akan didapat tegangan yang dapat terbaca oleh Arduino. Agar sensor tegangan ini dapat digunakan sebagai sensor, dibutuhkan program yang sesuai untuk sensor ini. *Flowchart* untuk program sensor tegangan ditunjukkan pada Gambar 3.11



Gambar 3.11 *Flowchart* Pemrograman Sensor Tegangan

Seperti yang telah digambarkan pada *Flowchart* diatas, untuk urutan cara kerja dari *flowchart* adalah sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Untuk *wiring* Sensor Tegangan dengan Arduino dihubungkan dengan pin analog Arduino atau pin ADC
3. Selanjutnya inisialisasi variabel yang akan digunakan untuk pembacaan dan penghitungan tegangan.
4. Selanjutnya membaca data pada analog Arduino atau ADC
5. Dengan menggunakan perhitungan sebagai berikut akan memperoleh nilai V_{rms} :

$$\text{Kalibrasi Tegangan} = \text{Tegangan Terukur} \dots\dots\dots(3.12)$$

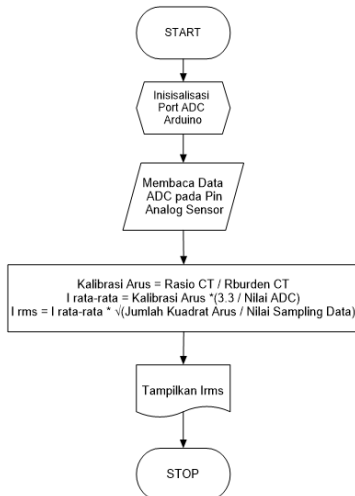
$$V_{rata-rata} = \text{Kalibrasi Tegangan} * \frac{3.3}{\text{Nilai ADC}} \dots\dots\dots(3.13)$$

$$V_{rms} = V_{rata-rata} * \sqrt{\frac{\text{Jumlah Kuadrat Arus}}{\text{Nilai Sampling Data}}} \quad (3.14)$$

6. Menampilkan nilai V_{rms}

3.4.2 Pemrograman Sensor Arus

Sensor arus yang digunakan pada Tugas Akhir ini adalah sensor arus *current transformers* SCT013-10 dimana *output* dari sensor ini berupa tegangan dan yang nantinya harus dapat terbaca oleh Arduino. Agar sensor arus ini dapat digunakan sebagai sensor, dibutuhkan program yang sesuai untuk sensor ini. *Flowchart* untuk program sensor arus ditunjukkan pada Gambar 3.12



Gambar 3.12 *Flowchart* Pemrograman Sensor Arus SCT-013

Seperti yang telah digambarkan pada *Flowchart* diatas, untuk urutan cara kerja dari *flowchart* adalah sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Untuk *wiring* Sensor Arus dengan Arduino dihubungkan dengan pin analog Arduino
3. Selanjutnya inisialisasi variabel yang akan digunakan untuk pembacaan dan penghitungan arus.
4. Selanjutnya membaca data pada analog Arduino atau ADC
5. Dengan menggunakan perhitungan sebagai berikut akan memperoleh nilai I_{rms} :

$$\text{Kalibrasi Arus} = \text{Rasio CT} / \text{Rburden CT} \dots\dots\dots(3.15)$$

$$I \text{ rata-rata} = \text{Kalibrasi Arus} * \frac{3.3}{\text{Nilai ADC}} \dots\dots\dots(3.16)$$

$$I_{rms} = I \text{ rata-rata} * \sqrt{\frac{\text{Jumlah Kuadrat Arus}}{\text{Nilai Sampling Data}}} \dots\dots\dots(3.17)$$

6. Menampilkan nilai I rms

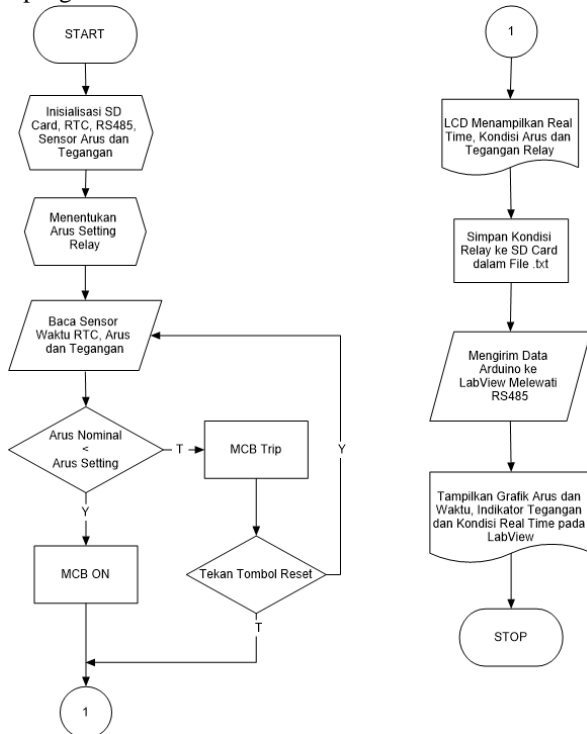
3.4.3 Pemrograman Relay

Relai ini berfungsi untuk mengamankan jaringan ketika sensor arus mendeteksi adanya arus lebih, sehingga relai ini memerintah MCB untuk memutuskan jaringan (*trip*). Selain itu relai ini juga dapat menampilkan status dari jaringan tersebut dan mengirimkannya ke data *logger* dan *human interface*. Agar relai dapat berjalan sesuai fungsinya, maka dibutuhkan program yang tepat pada papan Arduino.

Pertama, Arduino harus dapat mengenali SD Card, RTC, RS485, Sensor Arus dan Sensor Tegangan. Selanjutnya adalah menentukan arus *setting* relai. Arus *setting* ini digunakan untuk membatasi besar arus yang diperbolehkan untuk melewati jaringan. Lalu setelah mengenali sensor, Arduino akan membaca Sensor Arus, Tegangan dan Waktu RTC. Jika arus nominal atau arus yang terukur pada sensor kurang dari arus *setting* pada arduino, maka MCB pada relai akan *ON* dan menghubungkan jaringan (antara sumber dan beban). Namun jika sebaliknya, jika arus nominal atau arus yang terukur pada jaringan lebih besar dari arus *setting* maka MCB akan *OFF* atau *Trip*. Besarnya arus ini merupakan arus gangguan, dapat berupa karena beban lebih atau juga dapat berupa arus hubung singkat atau *short circuit*. Dengan *trip*-nya MCB ini, maka arus pada jaringan akan terputus sehingga antara sumber dan beban tidak terhubung. Atau dengan kata lain, jaringan atau sistem mati karena gangguan. Untuk mengembalikan agar jaringan tersambung lagi, maka perlu menekan tombol *reset* pada relai agar Arduino dapat membaca ulang arus dan tegangan yang melewati jaringan.

Kondisi yang terjadi pada relai ini akan ditampilkan pada LCD yang ada pada relai, sehingga *user* dapat langsung mengecek keadaan relai saat ini. Selanjutnya adalah menyimpan kondisi relai pada sebuah SD Card. Penyimpanan data ini berguna untuk Data *Logger* pada relai. Data yang ada pada arduino selanjutnya dikirimkan pada LabView melalui komunikasi RS485. Yang setelah itu data akan diproses pada LabView

dan menampilkan monitoring relai berupa grafik arus dan tegangan, waktu, indikator arus normal atau arus gangguan. Berikut ini adalah *Flowchart* dari program Arduino.



Gambar 3.13 *Flowchart* Pemrograman Relai Utama

BAB IV

PENGUJIAN DAN ANALISA

Untuk mengetahui kinerja dari peralatan dan pembuatan sistem yang telah dirancang dan direncanakan sedemikian rupa pada BAB III, maka diperlukan pengujian dan analisa data dari setiap komponen pendukung yang dibuat agar sistem dapat berjalan dengan baik sesuai dengan yang diharapkan. Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai pada pengujian sistem. Pengujian juga bertujuan untuk mengetahui kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan atau kesalahan dalam sistem

Pada bab ini akan dibahas tentang pengujian dan analisa data *hardware* dan *software* yang telah dibuat. Adapun bagian – bagian yang akan diuji pada alat ini adalah :

1. Pengujian UPS (*Uninterruptible Power Supply*)
2. Pengujian *Power Supply*
3. *Input/Output* Arduino
4. Pembacaan Sensor Arus
5. Pembacaan Sensor Tegangan
6. Pengujian Relai
7. Pengujian Koordinasi Adaptif Relai

4.1 Pengujian UPS (*Uninterruptible Power Supply*)

Pengujian UPS bertujuan untuk mengetahui besar tegangan pada UPS ketika dilakukan mode baterai dan saat diberi masukan 220 Volt. Pengujian ini dilakukan pada terminal *output* UPS. Pada Tugas Akhir ini, UPS dihubungkan pada *Power Supply* relai utama yang fungsinya sebagai sumber relai utama. Sehingga ketika jaringan terputus, relai utama tidak akan mati (kondisi *OFF*) yang menyebabkan berhenti melakukan pengukuran dan mencatat kondisi jaringan.

Pengukuran ini untuk mendapatkan besar arus, tegangan dan bentuk gelombang yang keluar dari UPS. Gambar 4.1 berikut ini adalah skema pengambilan data untuk UPS.



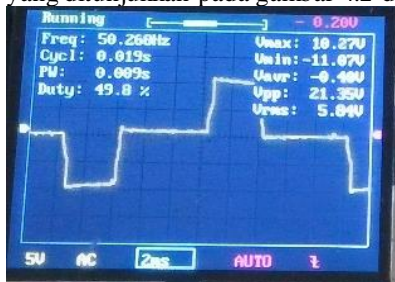
Gambar 4.1 Skema Pengujian UPS

Pengujian UPS ini dilakukan pada 2 kondisi, yaitu ketika tidak menggunakan sumber AC (sebagai baterai) dan ketika menggunakan *input* sumber AC. Berikut adalah hasil pengujian UPS yang digunakan untuk *input* Power Supply dan dapat dilihat pada tabel 4.1

Tabel 4.1 Pengujian Tegangan Keluaran UPS

No	Mode	Hasil
1	Baterai	153,4 V
2	Dengan <i>Input</i> 220 Volt	225,9 V

Untuk dapat membuktikan bahwa keluaran UPS merupakan gelombang AC, maka dibuktikan dengan melihat bentuk gelombang pada osiloskop. Sebelumnya, nilai tegangan *output* UPS dikecilkan terlebih dahulu dengan menghubungkan ke trafo *step-down* 220 / 5 Volt agar dapat dibaca oleh osiloskop. Berikut adalah bentuk gelombang dari output UPS yang ditunjukkan pada gambar 4.2 dan 4.3



Gambar 4.2 Bentuk Gelombang UPS sebagai Baterai



Gambar 4.3 Bentuk Gelombang UPS dengan *Input* 220 Volt

4.2 Pengujian Kecepatan Respon Sistem terhadap Halangan

Pengujian ini dilakukan pada *Power Supply* yang nantinya hasil tegangan keluarannya akan digunakan sebagai tegangan masukan Arduino. Tegangan masukan Arduino memiliki range 9-12 Volt DC dan *Power Supply* ini dapat mengubah tegangan 220 VAC menjadi 12 VDC. Pengujian ini menggunakan multimeter “SANWA CD800a” untuk mengukur besar tegangan *output Power Supply*. Pengukuran ini untuk mendapatkan besar arus, tegangan dan bentuk sinyal yang keluar dari *Power Supply* pada Relai Utama 1 dan Relai Utama 2. Gambar 4.1 berikut ini adalah skema pengambilan data untuk *Power Supply*.



Gambar 4.4 Skema Pengujian *Power Supply*

Setelah menyusun peralatan pengujian seperti skema diatas, dilakukan pengukuran arus dan tegangan pada *power supply* sebagai sumber arduino. Berikut adalah hasil pengujian *Power Supply* Relai Utama 1 dan Relai Utama 2 dapat dilihat pada tabel 4.1 dan 4.2

Tabel 4.2 Pengujian *Power Supply* Relai Utama 1

No	Parameter Pengujian	Hasil	Satuan
1	Tegangan	12,14	Volt
2	Arus	0,95	mA

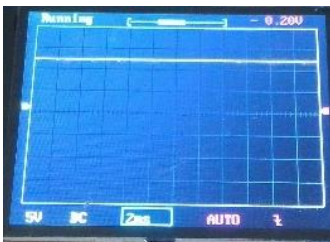
Tabel 4.3 Pengujian *Power Supply* Relai Utama 2

No	Parameter Pengujian	Hasil	Satuan
1	Tegangan	12,16	Volt
2	Arus	0,95	mA

Untuk dapat membuktikan bahwa keluaran *power supply* merupakan gelombang DC, maka dibuktikan dengan melihat bentuk gelombang pada osiloskop. Berikut adalah bentuk gelombang dari *power supply* yang ada pada Relai Utama 1 dan Relai Utama 2.



Gambar 4.5 Bentuk Gelombang *Power Supply* Relai Utama 1



Gambar 4.6 Bentuk Gelombang *Power Supply* Relai Utama 2

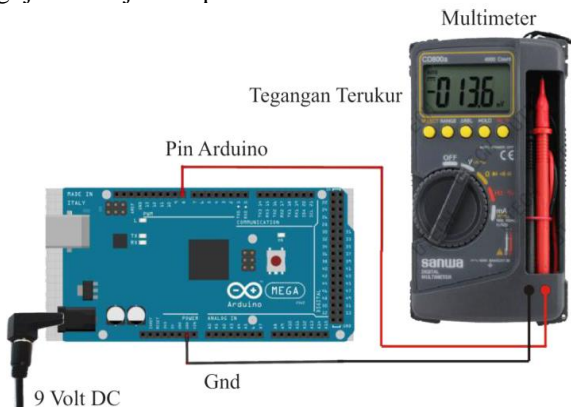
Dari data yang telah diambil, dapat disimpulkan bahwa tegangan yang keluar dari *Power Supply* sebesar 12 Volt dengan error sebesar 14-16 mV. Arus yang diperoleh sama yaitu sebesar 0,95 mA. Dan bentuk gelombang dari *output Power Supply* berupa DC murni. Berikut adalah cara pengambilan data *Power Supply* yang ditunjukkan oleh gambar 4.7



Gambar 4.7 Hasil Pengujian *Power Supply*.

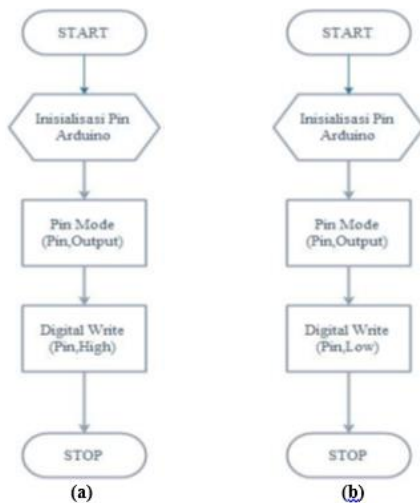
4.3 Pengujian *Input/Output Arduino*

Pegujian ini dilakukan terhadap *board* Arduino yang digunakan yakni Arduino Mega pada relai 1 dan Arduino Mega pada relai 2 . Pengujian ini bertujuan untuk mengetahui bahwa kondisi Arduino Mega kedua relai dapat digunakan dengan baik untuk Tugas Akhir ini. Skema pengujian ditunjukkan pada Gambar 4.8



Gambar 4.8 Skema Pengujian Pin *Input / Output* Arduino

Pengujian dilakukan dengan cara memberikan program pada Arduino yakni memberikan perintah *HIGH* dan *LOW* atau logika 1 dan 0 pada setiap pin Arduino yang akan diuji sesuai dengan *flowchart* yang ditunjukkan pada Gambar 4.9, kemudian mengukur besaran tegangan yang keluar dari pin tersebut seperti yang ditunjukkan pada Gambar 4.8. berikut adalah *flowchart* untuk pengujian tegangan pin Arduino Mega Dan Arduino Uno.



Gambar 4.9 Flowchart Program Pengujian Pin Arduino (a) Logic 1, (b) Logic 0

Pada pengujian pin arduino ini dibagi menjadi 2 pengujian. Yaitu untuk relai utama 1 dan relai utama 2. Berikut adalah hasil pengujian pin arduino :

Tabel 4.4 Pengujian Relai Utama 1

NO	NO PIN ARDUINO	LOGIC	TEGANGAN TERUKUR	LOGIC	TEGANGAN TERUKUR
1	A0	1	4.95 V	0	0.6 mv
2	A1	1	4.95 V	0	0.6 mv
3	A2	1	4.95 V	0	0.6 mv

4	A3	1	4.95 V	0	0.6 mv
5	A4	1	4.95 V	0	0.6 mv
6	A5	1	4.95 V	0	0.7 mv
7	D0	1	4.95 V	0	96.9 mv
8	D1	1	4.95 V	0	4.7 mv
9	D2	1	4.95 V	0	1.2 mv
10	D3	1	4.95 V	0	1.2 mv
11	D4	1	4.95 V	0	1.3 mv
12	D5	1	4.95 V	0	1.4 mv
13	D6	1	4.95 V	0	0.7 mv
14	D7	1	4.95 V	0	0.7 mv
15	D8	1	4.95 V	0	0.8 mv
16	D9	1	4.95 V	0	0.8 mv
17	D10	1	4.95 V	0	0.9 mv
18	D11	1	4.95 V	0	0.9 mv
19	D12	1	4.95 V	0	0.8 mv
20	D13	1	4.95 V	0	0.8 mv
21	D18	1	4.95 V	0	1.7 mv
22	D19	1	4.95 V	0	1.6 mv
23	D20	1	4.95 V	0	11.8 mv
24	D21	1	4.95 V	0	11.6 mv
25	D35	1	4.95 V	0	0.9 mv

26	D37	1	4.95 V	0	1.1 mv
27	D39	1	4.95 V	0	0.5 mv
28	D41	1	4.95 V	0	1.2 mv
29	D50	1	4.95 V	0	0.9 mv
30	D51	1	4.95 V	0	0.9 mv
31	D52	1	4.95 V	0	0.9 mv
32	D53	1	4.95 V	0	0.8 mv
V _{reff}		5V		4.95 V	
		VIN		11.39 V	
		3.3V		3.325 V	

Tabel 4.5 Pengujian Relai Utama 2

NO	NO PIN ARDUINO	LOGIC	TEGANGAN TERUKUR	LOGIC	TEGANGAN TERUKUR
1	A0	1	5.01 V	0	0.7 mv
2	A1	1	5.01 V	0	0.7 mv
3	A2	1	5.01 V	0	0.7 mv
4	A3	1	5.01 V	0	0.7 mv
5	A4	1	5.01 V	0	0.7 mv
6	A5	1	5.01 V	0	0.7 mv
7	D0	1	5.01 V	0	99.5 mv
8	D1	1	5.01 V	0	4.8 mv
9	D2	1	5.01 V	0	1.3 mv
10	D3	1	5.01 V	0	1.3 mv
11	D4	1	5.01 V	0	1.4 mv

12	D5	1	5.01 V	0	1.5 mv
13	D6	1	5.01 V	0	0.7 mv
14	D7	1	5.01 V	0	0.8 mv
15	D8	1	5.01 V	0	0.9 mv
16	D9	1	5.01 V	0	0.8 mv
17	D10	1	5.01 V	0	0.9 mv
18	D11	1	5.01 V	0	0.9 mv
19	D12	1	5.01 V	0	0.9 mv
20	D13	1	5.01 V	0	0.9 mv
21	D18	1	5.01 V	0	1.7 mv
22	D19	1	5.01 V	0	1.7 mv
23	D20	1	5.01 V	0	12.3 mv
24	D21	1	5.01 V	0	12.3 mv
25	D35	1	5.01 V	0	1.1 mv
26	D37	1	5.01 V	0	1.2 mv
27	D39	1	5.01 V	0	0.6 mv
28	D41	1	5.01 V	0	1.3 mv
29	D50	1	5.01 V	0	0.9 mv
30	D51	1	5.01 V	0	0.9 mv
31	D52	1	5.01 V	0	0.9 mv
32	D53	1	5.01 V	0	0.9 mv
V _{reff}		5V		5.01 V	
		VIN		11.41 V	
		3.3V		3.32 V	

4.4 Pembacaan Sensor Arus

Pada pengujian sensor arus ini kami melakukan kalibrasi sensor arus untuk mendapatkan tingkat akurasi pembacaan arus yang baik yaitu dengan mendapatkan nilai *error* pembacaan sekecil mungkin. Pengujian dilakukan kepada sensor arus SCT013-10 yang kami gunakan yang dengan menggunakan alat kalibrasi FLUKE 5500A. Berikut adalah gambar skema pengambilan data sensor arus.



Gambar 4.10 Skema Pengujian Sensor Arus

Pengujian sensor SCT013-10 dilakukan pada tiap-tiap relai, yaitu relai utama 1 dan relai utama 2. Nilai kalibrasi didapatkan dari sebuah persamaan dari pengujian awal sensor arus dengan menggunakan rumus *scatter* pada *Microsoft Excel*. Pada persamaan 4.1 dapat diketahui besar nilai *error* (%) yang ada. Berikut ini adalah pengujian untuk mendapatkan nilai persamaan kalibrasi.

$$Error = \left| \frac{Hasil\ Sensor - Hasil\ Alat\ Ukur}{Hasil\ Alat\ Ukur} \right| \times 100\% \dots\dots\dots (4.1)$$

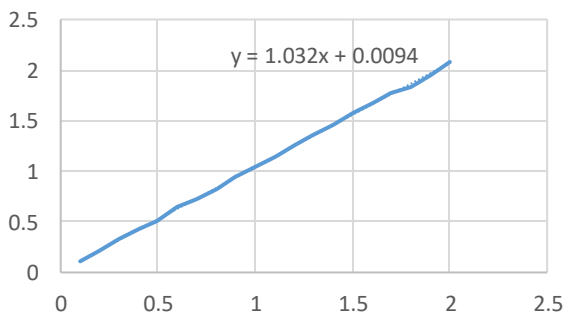
Tabel 4.6 Pengujian untuk Persamaan Kalibrasi Relai 1

KALIBRASI ARUS RELAY 1			ERROR (%)
NO	FLUKE 5500A	SCT013-10	
1	0,1	0,11	10,00
2	0,2	0,21	5,00
3	0,3	0,33	10,00
4	0,4	0,42	5,00
5	0,5	0,51	2,00
6	0,6	0,64	6,67

7	0,7	0,73	4,29
8	0,8	0,83	3,75
9	0,9	0,94	4,44
10	1	1,04	4,00
11	1,1	1,15	4,55
12	1,2	1,25	4,17
13	1,3	1,35	3,85
14	1,4	1,46	4,29
15	1,5	1,57	4,67
16	1,6	1,67	4,37
17	1,7	1,77	4,12
18	1,8	1,84	2,22
19	1,9	1,96	3,16
20	2	2,08	4,00
RATA-RATA ERROR (%)			4,73

Dari data diatas dapat diketahui grafik garis dan persamaannya. Berikut adalah hasil grafik garis beserta persamaannya :

PERSAMAAN KALIBRASI ARUS RELAI 1



Gambar 4.11 Grafik Garis dan Persamaan Kalibrasi Arus Relai 1

Untuk mendapatkan nilai kalibrasi, maka nilai awal kalibrasi pada program SCT013-10 di substitusikan ke persamaan garis. Karena nilai kalibrasi awal adalah 10, maka dapat dihitung sebagai berikut:

$$y = 1,032x + 0,0094 \dots\dots\dots (4.2)$$

$$10 = 1,032x + 0,0094 \dots\dots\dots (4.3)$$

$$x = 9.6808139534883720930232558139535$$

Setelah didapatkan nilai x, pada program sensor arus nilai kalibrasi diubah sesuai dengan nilai x. Maka akan didapatkan pengukuran sensor yang akurat.

Tabel 4.7 Pengujian untuk Data Arus Relai 1

DATA ARUS RELAY 1			ERROR (%)
NO	FLUKE 5500A	SCT013-10	
1	0,1	0,10	0,00
2	0,2	0,20	0,00
3	0,3	0,30	0,00
4	0,4	0,40	0,00
5	0,5	0,51	2,00
6	0,6	0,60	0,00
7	0,7	0,71	1,43
8	0,8	0,79	1,25
9	0,9	0,91	1,11
10	1	1,01	1,00
11	1,1	1,11	0,91
12	1,2	1,21	0,83
13	1,3	1,29	0,77
14	1,4	1,41	0,71
15	1,5	1,51	0,67
16	1,6	1,60	0,00
17	1,7	1,72	1,18

18	1,8	1,80	0,00
19	1,9	1,90	0,00
20	2	2,01	0,50
RATA-RATA ERROR (%)			0,62

Dapat dilihat pada rata-rata *error*, sebelum dan sesudah mendapat nilai kalibrasi yang benar. Pada saat sebelum mendapat nilai kalibrasi *error* sebesar 4,73%. Sedangkan sesudah mendapat nilai kalibrasi, *error* mengecil menjadi 0,42%. Untuk pengujian relai utama 2 juga dilakukan hal yang sama, dengan memberi nilai kalibrasi awal sebesar 10 dan nantinya akan diubah sesuai hasil yang didapatkan dari persamaan garis. Berikut adalah tabel persamaan kalibrasinya :

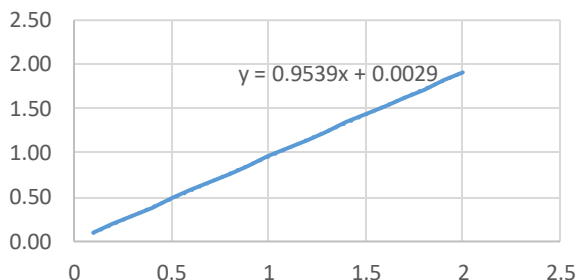
Tabel 4.8 Pengujian untuk Persamaan Kalibrasi Relai 2

KALIBRASI ARUS RELAY 2			
NO	FLUKE 5500A	SCT013-10	ERROR (%)
1	0,1	0,10	0,00
2	0,2	0,19	5,00
3	0,3	0,29	3,33
4	0,4	0,38	5,00
5	0,5	0,48	4,00
6	0,6	0,58	3,33
7	0,7	0,67	4,29
8	0,8	0,77	3,75
9	0,9	0,86	4,44
10	1	0,96	4,00
11	1,1	1,05	4,55
12	1,2	1,15	4,17
13	1,3	1,24	4,62

14	1,4	1,34	4,29
15	1,5	1,43	4,67
16	1,6	1,53	4,38
17	1,7	1,62	4,71
18	1,8	1,72	4,44
19	1,9	1,82	4,21
20	2	1,91	4,50
RATA-RATA ERROR			4,08

Dari data diatas dapat diketahui grafik garis dan persamaannya. Berikut adalah hasil grafik garis beserta persamaannya :

PERSAMAAN KALIBRASI ARUS RELAI 2



Gambar 4.12 Grafik Garis dan Persamaan Kalibrasi Arus Relai 2

Untuk mendapatkan nilai kalibrasi, maka nilai awal kalibrasi pada program SCT013-10 di substitusikan ke persamaan garis. Karena nilai kalibrasi awal adalah 10, maka dapat dihitung sebagai berikut:

$$y = 0,9539x + 0,0029 \dots\dots\dots (4.4)$$

$$10 = 0,9539x + 0,0029 \dots\dots\dots (4.5)$$

$$x = 10.480239018765069713806478666527$$

Setelah didapatkan nilai x, pada program sensor arus nilai kalibrasi diubah sesuai dengan nilai x. Maka akan didapatkan pengukuran sensor yang akurat.

Tabel 4.9 Pengujian untuk Data Arus Relai 2

DATA ARUS RELAY 2			ERROR (%)
NO	FLUKE 5500A	SCT013-10	
1	0,1	0,10	0,00
2	0,2	0,20	0,00
3	0,3	0,30	0,00
4	0,4	0,40	0,00
5	0,5	0,50	0,00
6	0,6	0,60	0,00
7	0,7	0,70	0,00
8	0,8	0,80	0,00
9	0,9	0,90	0,00
10	1	1,00	0,00
11	1,1	1,10	0,00
12	1,2	1,20	0,00
13	1,3	1,30	0,00
14	1,4	1,40	0,00
15	1,5	1,50	0,00
16	1,6	1,60	0,00
17	1,7	1,70	0,00
18	1,8	1,80	0,00
19	1,9	1,90	0,00
20	2	2,00	0,00
RATA-RATA ERROR			0,00

Dapat dilihat pada rata-rata *error*, sebelum dan sesudah mendapat nilai kalibrasi yang benar. Pada saat sebelum mendapat nilai kalibrasi *error* sebesar 4,08%. Sedangkan sesudah mendapat nilai kalibrasi, *error*

mengecil hingga mendekati 0,00% atau tidak terdapat *error*. Dengan begitu, sensor arus ini sudah dapat digunakan dengan baik, karena tidak memiliki nilai *error* yang besar. Berikut adalah proses pengujian sensor arus



Gambar 4.13 Proses Pengujian Sensor Arus

4.5 Pembacaan Sensor Tegangan

Pada pengujian sensor tegangan ini kami melakukan kalibrasi sensor tegangan untuk mendapatkan tingkat akurasi pembacaan arus sebaik mungkin. Pengujian dilakukan kepada trafo CT 350 mA yang kami gunakan yang dengan menggunakan alat kalibrasi FLUKE 5500A. Berikut adalah gambar skema pengambilan data sensor arus.



Gambar 4.14 Skema Pengujian Sensor Tegangan

Pengujian sensor tegangan ini dilakukan pada tiap-tiap relai, yaitu relai utama 1 dan relai utama 2. Dengan persamaan 4.1 dapat diketahui besarnilai *error* (%) yang ada. Berikut ini adalah pengujian untuk sensor tegangan relai 1 dan relai 2.

Tabel 4.10 Pengujian untuk Data Tegangan Relai 1

DATA SENSOR TEGANGAN RELAY 1			
NO	FLUKE 5500A	SCT013-10	ERROR(%)
1	100	100,62	0,62
2	110	110,72	0,65
3	120	120,74	0,62
4	130	130,77	0,59
5	140	140,98	0,70
6	150	150,40	0,27
7	160	161,19	0,74
8	170	170,99	0,58
9	180	181,12	0,62
10	190	191,04	0,55
11	200	201,25	0,63
12	210	211,25	0,60
13	220	221,28	0,58
14	230	231,14	0,50
15	240	240,43	0,18
16	250	250,78	0,31
RATA-RATA ERROR			0,55

Tabel 4.11 Pengujian untuk Data Tegangan Relai 2

DATA SENSOR TEGANGAN RELAY 2			
NO	FLUKE 5500A	SCT013-10	ERROR (%)
1	100	100,24	0,24
2	110	110,12	0,11
3	120	120,30	0,25

4	130	130,52	0,40
5	140	140,39	0,28
6	150	150,63	0,42
7	160	160,84	0,53
8	170	170,86	0,51
9	180	180,97	0,54
10	190	190,86	0,45
11	200	201,31	0,66
12	210	211,21	0,58
13	220	221,70	0,77
14	230	231,42	0,62
15	240	241,63	0,68
16	250	251,03	0,41
RATA-RATA ERROR			0,46

Dari data yang telah diperoleh dari pengujian sensor tegangan, dapat diketahui bahwa nilai *error* sensor tegangan pada relai 1 sebesar 0,55% dan pada relai 2 sebesar 0,46%. Dengan begitu, sensor tegangan ini dapat digunakan dengan baik, karena tidak memiliki nilai *error* yang besar.

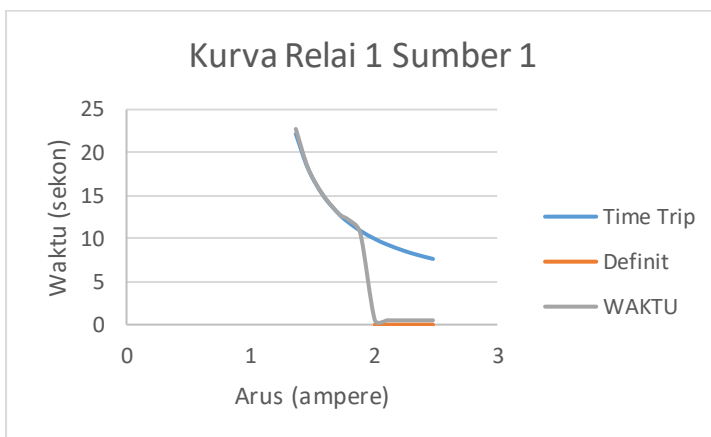
4.6 Pengujian Relay

Pada Pengujian relai ini kami memasang relai pada sumber tegangan satu fasa dan memberikan nilai *setting overload* sebesar 1A, *shortcircuit* sebesar 2A dan *time dial* sebesar 1s pada masing-masing relai. Relai tersebut nantinya akan diberikan beban untuk mendeteksi apakah relai berfungsi sebagaimana fungsinya yaitu untuk memutus saluran tenaga listrik apabila terjadi kenaikan arus melebihi arus *setting over load* dan *short circuit test*. Berikut merupakan tabel pengujian relai.

Tabel 4.12 Pengujian Relai 1 dengan 1 sumber

NO	BEBAN			I NOM	Time Trip	Definit	WAKTU	KONDISI
1	rice cooker 300w		-	1.37	22.1657		22.75	OL
2	rice cooker 300w	lampu 25 w	-	1.47	18.0995		18.17	OL
3	rice cooker 300w	lampu 50 w	-	1.59	15.025		15.09	OL
4	rice cooker 300w	lampu 75 w	-	1.72	12.8375		12.86	OL
5	rice cooker 300w	lampu 100 w	-	1.79	11.9532		12.25	OL
6	rice cooker 300w	lampu 100 w	lampu 25 w	1.89	10.9265		10.72	OL
7	rice cooker 300w	lampu 100 w	lampu 50 w	2.01	9.95688	0	0.5	SC
8	rice cooker 300w	lampu 100 w	lampu 75 w	2.11	9.30491	0	0.5	SC
9	rice cooker 300w	lampu 100 w	lampu 100 w	2.24	8.60993	0	0.5	SC
10	impedansi	lampu 25 w	-	2.35	8.12294	0	0.5	SC
11	impedansi	lampu 50 w	-	2.48	7.63727	0	0.5	SC

*Ket OL= Overload, SC = Shortcircuit.

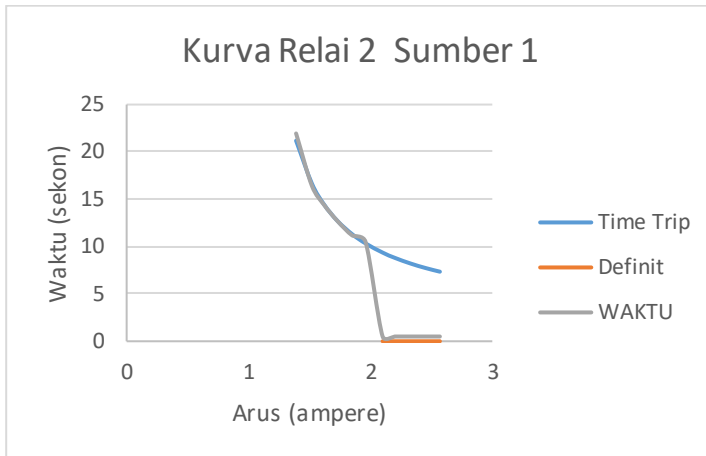


Gambar 4.15 Kurva Kerja Relai 1 Pada Sumber 1

Tabel 4.13 Pengujian Relai 2 dengan 1 sumber

NO	BEBAN			I NOM	Time Trip	Definit	WAKTU	KONDISI
1	rice cooker 300w			1.39	21.187		21.95	OL
2	rice cooker 300w	lampu 25 w		1.52	16.6481		16.41	OL
3	rice cooker 300w	lampu 50 w		1.62	14.4401		14.4	OL
4	rice cooker 300w	lampu 75 w		1.73	12.701		12.69	OL
5	rice cooker 300w	lampu 100 w		1.85	11.3088		11.18	OL
6	rice cooker 300w	lampu 100 w	lampu 25 w	1.96	10.3322		10.5	OL
7	rice cooker 300w	lampu 100 w	lampu 50 w	2.1	9.36493	0	0.5	SC
8	rice cooker 300w	lampu 100 w	lampu 75 w	2.2	8.80828	0	0.5	SC
9	rice cooker 300w	lampu 100 w	lampu 100 w	2.33	8.20572	0	0.5	SC
10	impedansi	lampu 25 w		2.45	7.74194	0	0.5	SC
11	impedansi	lampu 50 w		2.57	7.34621	0	0.5	SC

*Ket OL= Overload, SC = Shortcircuit.



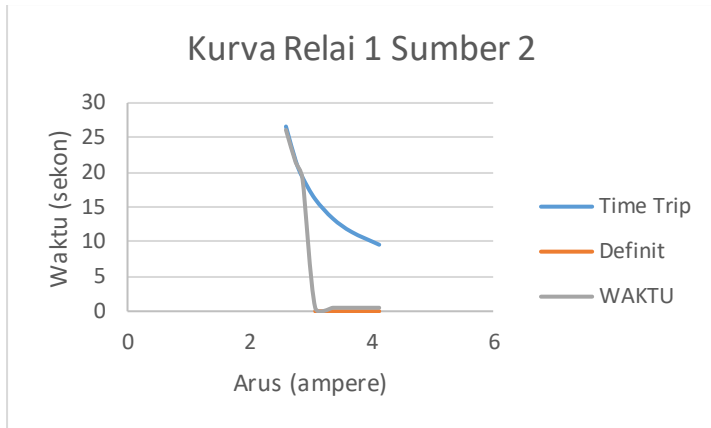
Gambar 4.16 Kurva Kerja Relai 2 Pada Sumber 1 .

Dari data tabel 4.12 dan 4.13 yang merupakan tabel hasil pengujian relai 1 dan relai 2, maka dapat dibentuk suatu kurva seperti gambar 4.15 dan gambar 4.16, dimana kurva tersebut memiliki bentuk sesuai setting relai yang mengacu pada fungsi campuran, yaitu fungsi “*Standard Inverse*” untuk arus overload dan fungsi “*Devinite Time*” untuk arus *Short Circuit*.

Tabel 4.14 Pengujian Relai 1 dengan 2 sumber

NO	BEBAN			I NOM	Time Trip	Definit	WAKTU	KONDISI
1	Setrika	Rice Cooker 30w	lampu 175 w	2.6	26.6105		26.11	OL
2	Setrika	Rice Cooker 30w	lampu 200 w	2.76	21.6636		21.42	OL
3	Setrika	Rice Cooker 30w	lampu 225 w	2.87	19.3118		19.15	OL
4	Setrika	Rice Cooker 300w		3.08	16.142	0	0.5	SC
5	Impedansi		lampu 225 w	3.37	13.3461	0	0.5	SC
6	Impedansi	Rice Cooker 300w		3.68	11.41	0	0.5	SC
7	Impedansi	Rice Cooker 300w	lampu 100 w	4.13	9.58361	0	0.5	SC

*Ket OL= Overload, SC = Shortcircuit.

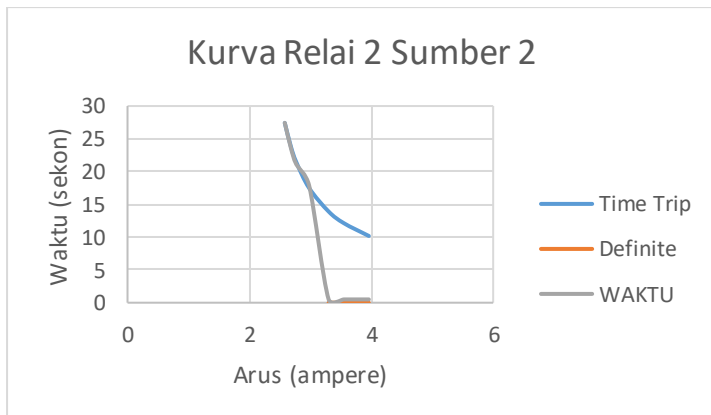


Gambar 4.17 Kurva Kerja Relai 1 Pada Sumber 2

Tabel 4.15 Pengujian Relai 2 dengan 2 sumber

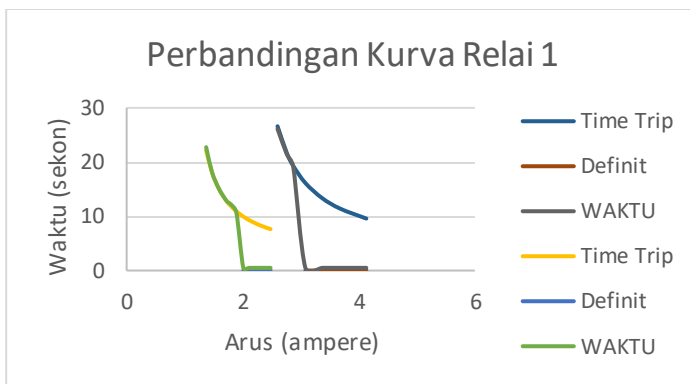
NO	BEBAN			I NOM	Time Trip	Definite	WAKTU	KONDISI
1	Setrika	Rice Cooker 30w	lampu 175 w	2.58	27.4196		27.44	OL
2	Setrika	Rice Cooker 30w	lampu 200 w	2.68	23.8478		23.62	OL
3	Setrika	Rice Cooker 30w	lampu 225 w	2.77	21.4222		21.09	OL
4	Setrika	Rice Cooker 300w		2.98	17.4838		17.86	OL
5	Impedansi		lampu 225 w	3.3	13.9084	0	0.5	SC
6	Impedansi	Rice Cooker 300w		3.55	12.1295	0	0.5	SC
7	Impedansi	Rice Cooker 300w	lampu 100 w	3.96	10.1776	0	0.5	SC

*Ket OL= Overload, SC = Shortcircuit.

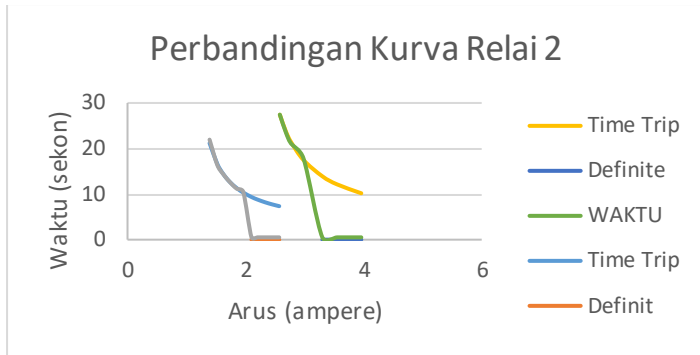


Gambar 4.18 Kurva Kerja Relai 1 Pada Sumber 2

Sama halnya pengujian relai dengan 1 sumber, Data dari tabel 4.14 dan 4.15 yang merupakan hasil pengujian relai 1 dan relai 2, maka dapat dibentuk suatu kurva seperti gambar 4.17 dan gambar 4.18, dimana kurva tersebut memiliki bentuk sesuai setting relai yang mengacu pada fungsi campuran, yaitu fungsi “*Standard Inverse*” untuk arus *overload* dan fungsi “*Devinite Time*” untuk arus *shortcircuit*. Namun perbedaan terlihat pada besar arus *setting overload* dan *shortcircuit*, dimana besarnya nilai arus *setting* bertambah seiringan dengan adanya penambahan sumber pada sistem secara otomatis. Hal ini mengacu pada sifat relai yang bersifat adaptif terhadap penambahan sumber. Berikut adalah kurva perbandingan perubahan setting relai dari yang awalnya hanya menggunakan 1 sumber berubah ketika terjadi penambahan sumber.



Gambar 4.19 Kurva Perbandingan Setting Relai 1



Gambar 4.20 Kurva Perbandingan Setting Relai 2

4.7 Pengujian Koordinasi Adaptif Relay

Pengujian koordinasi adaptif relai ini bertujuan untuk menentukan nilai-nilai dari settingan waktu kerja (MCB trip) dan arus kerja dari suatu alat proteksi relai yang dipasang pada sistem agar dapat bekerja terorganisir pada saat sistem bekerja. Pengujian ini dilakukan dengan memasang relai pada bus 1 dan bus 2 yang kemudian akan diberikan beban pada tiap-tiap bus hingga melebihi nilai arus *setting overload* maupun *shortcircuit*. Pada saat melakukan pengujian koordinasi relai, penguji mengambil 3 kondisi, yaitu :

1. Kondisi Arus *setting* sama dan *time dial* berbeda
2. Kondisi Arus *setting* berbeda dan *time dial* sama
3. Kondisi Arus *setting* berbeda dan *time dial* berbeda

Berikut adalah gambar dan tabel pengujian koordinasi adaptif relai pada 1 sumber dan 2 sumber.



Gambar 4.21 Proses Pengujian Koordinasi Adaptif Relai

Tabel 4.16 Hasil Pengujian pada Koordinasi Relai 1 Sumber

1 SUMBER													
KOORDINASI OVERLOAD													
No.	Time Dial Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip		
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2	
1	1	1	1,3	1	0	425 Watt	1.94 A	1.9 A	OK	Trip	-	10.83 s	
2	1	1	1,3	1	425 Watt	0	1.96 A	0.04 A	Trip	OK	16.49 s	-	
3	1	1	1,3	1	300 Watt	100 Watt	1.87 A	0.44 A	Trip	OK	19.93 s	-	
4	1	1	1,3	1	100 Watt	300 Watt	1.88 A	1.37 A	Trip	OK	20.24 s	-	
5	1,2	1	1	1	0	425 Watt	1.94 A	1.88 A	OK	Trip	-	11.88 s	
6	1,2	1	1	1	425 Watt	0	1.94 A	0.04 A	Trip	OK	12.60 s	-	
7	1,2	1	1	1	300 Watt	100 Watt	1.74 A	0.44 A	Trip	OK	13.73 s	-	
8	1,2	1	1	1	100 Watt	300 Watt	1.82 A	1.33 A	Trip	OK	14.69 s	-	
9	1,2	1	1,3	1	100 Watt	300 Watt	1.82 A	1.33 A	OK	Trip	-	23.81 s	

SHORT CIRCUIT													
No.	Time Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip		
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2	
1	2	1	2	2	0	Shortcircuit	2.29 A	2.29 A	OK	Trip	-	1 s	
2	2	1	2	2	Shortcircuit	Load	2.29 A	1.7 A	Trip	OK	2 s	-	
3	1	1	2,3	2	0	Shortcircuit	0	2.2 A	OK	Trip	-	2 Detik	
4	1	1	2,3	2	Shortcircuit	Load	2.5 A	1.5 A	Trip	OK	1 s	-	

Tabel 4.17 Hasil Pengujian pada Koordinasi Relai 2 Sumber

2 SUMBER

KOORDINASI OVERLOAD

No.	Time Dial Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip	
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2
1	1	1	2,3	2	0	575 Watt	2.6 A	2.6 A	OK	Trip	-	27,41 s
2	1	1	2,3	2	575 Watt	0	2.7 A	0.04 A	Trip	OK	43.66 s	-
3	1	1	2,3	2	550 Watt	100 Watt	2.85 A	0.4 A	Trip	OK	33.15 s	-
4	1	1	2,3	2	100 Watt	550 Watt	2.84 A	2.48 A	Trip	OK	32.40 s	-
5	1,2	1	2	2	0	575 Watt	2,54	2.54 A	OK	Trip	-	31,51 s
6	1,2	1	2	2	575 Watt	0	2.57 A	0.03 A	Trip	OK	33.45 s	-
7	1,2	1	2	2	550 Watt	100 Watt	2.9 A	0.44 A	Trip	OK	22.6 s	-
8	1,2	1	2	2	100 Watt	500 Watt	2,91	2.47 A	Trip	OK	22.8 s	-
9	1,2	1	2,3	2	100 Watt	500 Watt	2.9 A	2.47 A	OK	Trip	-	32.94 s

SHORT CIRCUIT

No.	Time Set		I nominal Set		Daya Beban		Arus Nominal		Kondisi Relay		Waktu Trip	
	Relay 1	Relay 2	Relay 1	Relay 2	Bus 1	Bus 2	Relay 1	Relay 2	Relay 1	Relay 2	Relay 1	Relay 2
1	2	1	3	3	0	Shortcircuit	0	3.5 A	OK	Trip	-	1 s
2	2	1	3	3	Shortcircuit	Load	6.1 A	2.9 A	Trip	OK	2 s	-
3	1	1	3,3	3	0	Shortcircuit	3,56	3,56	Trip	Trip	1 s	1 s
4	1	1	3,3	3	Shortcircuit	Load	3.5 A	2.5 A	Trip	OK	1 s	-

Dari data pada tabel diatas, dapat dilihat bahwa koordinasi antar *relay* sangat dipengaruhi oleh besarnya *setting* pada *time dial*, arus *Overload*, dan *Shortcircuit*. Oleh karena itu dalam pengamanan system proteksi tenaga listrik haruslah diiringi dengan pengoordinasian yang tepat, guna menjaga kehandalan sistem proteksi tersebut.

4.8 Analisa Relevansi

Alat ini merupakan perancangan miniatur dari *Over Current Relay* yang nantinya dapat digunakan sebagai pembelajaran dalam sistem proteksi tenaga listrik. Namun dalam proses pembelajaran menggunakan alat ini perlu dilakukan penyesuaian terhadap dengan beban yang akan diukur, karena pada alat ini kemampuan pengukuran hanya dibatasi pada beban dengan batas arus maksimal 4 A dan hanya dapat digunakan untuk beban 1 fasa. Untuk melakukan pengukuran pada beban 3 fasa maka perlu dilakukan penambahan komponen pada alat yang telah dibuat pada Tugas Akhir ini.

Diharapkan dengan adanya miniatur dari *Over Current Relay* maka dapat mempermudah proses pembelajaran dan analisa dari sistem proteksi pada jaringan satu fasa. Karena dengan alat ini akan dapat disimulasikan langsung bagaimana proses dalam sistem proteksi yang biasanya dipakai di PT PLN(Persero).

BAB V PENUTUP

Bab penutup ini berisi kesimpulan yang diperoleh selama proses pembuatan tugas akhir yang berjudul **“Desain Prototipe Dan Koordinasi Adaptif Relai Arus Lebih Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa”**, serta saran untuk pengembangan alat ini kedepannya.

5.1 Kesimpulan

Berdasarkan hasil pengujian pada Tugas Akhir ini, maka diperoleh beberapa kesimpulan yaitu :

1. Dalam pembuatan prototipe adaptif relai arus lebih, hal terpenting yang perlu diperhatikan yaitu kalibrasi sensor arus dan tegangan, dimana kedua hasil pengukuran sensor ini akan digunakan untuk menentukan nilai setting relai. Oleh karena itu diperlukan nilai kalibrasi pengukuran yang nilai *error*-nya mendekati atau sama dengan 0,00%. Ditunjukkan pada tabel 4.7 dan tabel 4.9 bahwa nilai pengukuran arus memiliki *error* sebesar 0,62% pada relai 1 dan 0,00% pada relai 2
2. Ketika terjadi penambahan sumber relai dapat merubah nilai *setting* arus gangguan secara otomatis, hal ini dibuktikan pada gambar 4.19 dan gambar 4.20 dimana bentuk kurva *setting* relai berubah saat terjadi penambahan sumber.
3. Relai mampu berkoordinasi dengan baik dan dapat mengamankan saluran listrik satu fasa dari gangguan *overload* dan *short circuit*. Hal ini dibuktikan dengan tabel 4.16 dan tabel 4.17 bahwa pengoordinasian yang baik ditentukan dengan nilai *setting* arus gangguan dan *setting time dial* pada masing-masing relai.

5.2 Saran

Saran untuk penelitian selanjutnya adalah penambahan karakteristik *setting* relai seperti karakteristik *very inverse*, *short time inverse*, *longtime inverse*, dan *extremely inverse*. Selain itu juga perlu ditambahkan beberapa variasi pengujian lain pada simulasi maupun implementasi guna pembelajaran terhadap analisa sistem proteksi tenaga listrik.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1], **PENGESAHAN RUPTL PLN TAHUN 2016-2025**, MENTERI ENERGI DAN SUMBER DAYA MINERAL REPUBLIK INDONESIA, 2016
- [2] Andrianto, Heri; dan Aan Darmawan. 2016. **Arduino Belajar Cepat dan Pemrograman**. Bandung : INFORMATIKA.
- [3], **IDE Software for Arduino**, Datasheet, 2015.
- [4], **INFORCE UNINTERRUPTIBLE POWER SUPPLY 650 VA**, Manual Book, 2009.
- [5] DFROBOT, **ARDUINO LCD KEYPAD DFROBOT**, <https://www.dfrobot.com/>, 19 Mei 2017.
- [6], **OPEN ENERGY MONITOR**, <https://openenergymonitor.org/>, 15 April 2017
- [7] **AZZETTLER RELAYS**, <http://www.azettler.com>, 16 April 2017
- [8] Adhitya, W.W. dan Faisal, A., **PERANCANGAN SISTEM MONITORING VOLTAGE FLICKER BERBASIS ARDUINO DENGAN METODE FAST FOURIER TRANSFORM (FFT)**, Tugas Akhir, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2016.
- [9], **BUKU PANDUAN PUSDIKLAT PLN BAB VII, RELAI ARUS LEBIH**, 2016
- [10], **BUKU PANDUAN PUSDIKLAT PLN BAB IX, KOORDINASI PROTEKSI DISTRIBUSI**, 2016

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1 Listing Program Relai Indikator (Master)

```
#include <SoftwareSerial.h>
#define SSerialRX      11 //Serial Receive pin
#define SSerialTX      10 //Serial Transmit pin

#define SSerialTxControl 9 //RS485 Direction control

#define RS485Transmit   HIGH
#define RS485Receive    LOW

#define PinLED          13

/*-----( Declare objects )-----*/
SoftwareSerial RS485Serial(SSerialRX, SSerialTX); // RX, TX

const int buttonPin1 = 4; // the number of the pushbutton pin
const int ledPin1 = 5; // the number of the LED pin
const int buttonPin2 = 7; // the number of the pushbutton pin
const int ledPin2 = 6; // the number of the LED pin

// variables will change:
int buttonState1=0; // variable for reading the pushbutton status
int buttonState2= 0;
/*-----( Declare Variables )-----*/
int byteReceived;
int byteSend;
int data;

void setup()
{
  Serial.begin(9600);

  pinMode(PinLED, OUTPUT);
  pinMode(SSerialTxControl, OUTPUT);

  digitalWrite(SSerialTxControl, RS485Receive); // Init
  Transceiver
```

```

// Start the software serial port, to another device
RS485Serial.begin(9600); // set the data rate
// initialize the LED pin as an output:
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
// initialize the pushbutton pin as an input:
pinMode(buttonPin1, INPUT);
pinMode(buttonPin2, INPUT);
}

void loop()
{
buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
// check if the pushbutton is pressed.
// if it is, the buttonState is HIGH:

if (buttonState1 == HIGH && buttonState2 == LOW)
{
digitalWrite(ledPin1, HIGH);
delay(1000);
digitalWrite(SSerialTxControl, RS485Transmit); // Enable
RS485 Transmit
byteSend = RS485Serial.write("a"); // Send byte to Remote
Arduino

digitalWrite(PinLED, LOW); // Show activity
digitalWrite(SSerialTxControl, RS485Receive); // Disable
RS485 Transmit
}
else {
// turn LED off:
digitalWrite(ledPin1, LOW);
}
if (buttonState2 == HIGH && buttonState1 == LOW) {
// turn LED on:
digitalWrite(ledPin2, HIGH);
delay(1000);
}
}

```

```

    digitalWrite(SSerialTxControl, RS485Transmit);    // Enable
RS485 Transmit
    byteSend = RS485Serial.write("a");           // Send byte to Remote
Arduino

    digitalWrite(PinLED, LOW); // Show activity
    digitalWrite(SSerialTxControl, RS485Receive);    // Disable
RS485
}
else {
    // turn LED off:
    digitalWrite(ledPin2, LOW);
}
if (buttonState1 == HIGH && buttonState2 == HIGH ) {
    // turn LED on:
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, HIGH);
    delay(1000);
    digitalWrite(SSerialTxControl, RS485Transmit);    // Enable
RS485 Transmit
    byteSend = RS485Serial.write("b");           // Send byte to Remote
Arduino

    digitalWrite(PinLED, LOW); // Show activity
    digitalWrite(SSerialTxControl, RS485Receive);    // Disable
RS485
}
else {
    // turn LED off:
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
}
}
}

```

A.2 Listing Program Relai Utama (Slave)

```

#if ARDUINO < 100
#include <WProgram.h>
#else
#include <Arduino.h>

```

```

#endif

void setup (){
    setupRelayShortCirt();
    setupRelayOverLoad();
    setupDefault();
    pinMode(pinCS,OUTPUT);
    Serial.begin(9600);

void setting(){
    if (Serial1.available()>0)
    {
        data = Serial1.read(); // Read the byte
        //Serial.println(data);
        if (data == 97){
            // put your main code here, to run repeatedly:
            setupDefault();
        }
        if (data == 98){
            // put your main code here, to run repeatedly:
            setupDefault2();
        }
    }
}

//Kalau Sumber 1
void setupDefault(){
    Ishortcircuit = 2.3;
    Ioverload =1.3;
    /*setup Relay Protection Short Circuit*/
    RelayShortCirt.setTimeChar(DEF_TIME);//Time Characteristic
    //RelayShortCirt.setTimeChar(INST_TIME);//Time Characteristic
    //RelayShortCirt.setTimeChar(INVS_STD);//Time Characteristic
    RelayShortCirt.setTimeDef(1000);
    RelayShortCirt.setSetting(Ishortcircuit);//set setting value
    //RelayShortCirt.setTMS(1);//set setting value
    /*setup Relay Protection Over Load (Tambahan 20/05/2017)*/
    //RelayOverLoad.setTimeChar(DEF_TIME);//Time Characteristic
    //RelayOverLoad.setTimeChar(INST_TIME);//Time
    Characteristic

```

```

RelayOverLoad.setTimeChar(INVS_STD);//Time Characteristic
//RelayOverLoad.setTimeDef(2000);
RelayOverLoad.setSetting(Ioverload);//set setting value
RelayOverLoad.setTMS(1200);//set setting value

}
//Kalau Sumber 2
void setupDefault2(){
  Ishortcircuit = 3.3;
  Ioverload =2.3;
  /*setup Relay Protection Short Circuit*/
  RelayShortCirt.setTimeChar(DEF_TIME);//Time Characteristic
  //RelayShortCirt.setTimeChar(INST_TIME);//Time Characteristic
  //RelayShortCirt.setTimeChar(INVS_STD);//Time Characteristic
  RelayShortCirt.setTimeDef(1000);
  RelayShortCirt.setSetting(Ishortcircuit);//set setting value
  //RelayShortCirt.setTMS(1);//set setting value
  /*setup Relay Protection Over Load (Tambahan 20/05/2017)*/
  //RelayOverLoad.setTimeChar(DEF_TIME);//Time Characteristic
  //RelayOverLoad.setTimeChar(INST_TIME);//Time
Characteristic
  RelayOverLoad.setTimeChar(INVS_STD);//Time Characteristic
  //RelayOverLoad.setTimeDef(2000);
  RelayOverLoad.setSetting(Ioverload);//set setting value
  RelayOverLoad.setTMS(1200);//set setting value
}

void setupRelayShortCirt(){
  //setup Relay Protection Short Circuit
  RelayShortCirt.setPinTest(PIN_TRIP_SWITCH);
  RelayShortCirt.setPinReset(PIN_RESET_SWITCH);
  //RelayShortCirt.setPinLED(LED_SH);
  RelayShortCirt.setPinRL(PIN_RELAY_OUT);
  RelayShortCirt.setActionChar(OVR_ACTION);
  RelayShortCirt.setActive(true);

void setupRelayOverLoad(){
  //setup Relay Protection Over Load
  RelayOverLoad.setPinTest(PIN_TRIP_SWITCH);

```

```

RelayOverLoad.setPinReset(PIN_RESET_SWITCH);
//RelayOverLoad.setPinLED(LED_SH);
RelayOverLoad.setPinRL(PIN_RELAY_OUT);
RelayOverLoad.setActionChar(OVR_ACTION);
RelayOverLoad.setActive(true);

```

A.3 Listing Program Sensor Arus dan Tegangan

```

#include "EmonLib.h"
EnergyMonitor monitoringArusDanTegangan;

void setup() {
    Serial.begin(9600);
    SetupEmonlib();

void loop() {
    if (SequenceUtama.isMiliSecondEvent()) { //Dijalankan Tiap 500
ms
        //updateCommandFromHost();//check data dari Host dulu dan
transfer sebagian ke operasi
        setting();
        transferFromProcessToHoldingRegsArdu();//
        mainMenuRele();
    }
    if (SequenceUtama.isASecondEvent()){//jalan setiap detik
//Baca Data Dari EmonLib (Letakkan Program dibawah)

monitoringArusDanTegangan.calcVI(CROSSING_MAX,TIME_O
UT_MAX); // Calculate all. No.of half wavelengths (crossings),
time-out
        digitalWrite( PIN_LED_LIVE, digitalRead(PIN_LED_LIVE) ^
1 );
        SaatIni = rtc.now();
    }

void SetupEmonlib(){
    //Inisialisasi EmonLib Tegangan dan Arus
    monitoringArusDanTegangan.voltage(1, 240, 1.7); // Voltage:
input pin, calibration, phase_shift

```



```

    //monitoringArusDanTegangan.voltage(1, 250, 1.7); // Voltage:
input pin, calibration, phase_shift
    monitoringArusDanTegangan.current(3, 10.42); // Current: input
pin, calibration.
    //monitoringArusDanTegangan.current(3,
9.6808139534883720930232558139535); // Current: input pin,
calibration. (RL1)
    //monitoringArusDanTegangan.current(3,
10.480239018765069713806478666527); // Current: input pin,
calibration. (RL2)
}

```

A.4 Listing Program MMC (SD Card) dan RTC

```

// Program penyimpanan memori Micro SD pada Tanggal dan
Waktu menggunakan DS1307 RTC terkoneksi dalam I2C dan
Wire lib
#include <Wire.h>
#include "RTClib.h"
#include <SD.h>

```

```

File myFile; //Pendeklarasian File
RTC_DS1307 rtc; //Pendeklarasian RTC 1307

```

```

char Time[20]; //Pendeklarasian Waktu
char Date[20]; //Pendeklarasian Tanggal
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};
int pinCS = 53; // Gunakan pin 10 pada Arduino Uno atau pin 53
pada Arduino Mega

```

```

void setup() {

    Serial.begin(9600);
    pinMode(pinCS, OUTPUT);
    // Inisialisasi SD Card
    if (SD.begin())
    {
        Serial.println("SD card is ready to use.");
    } else

```

```

{
  Serial.println("SD card initialization failed");
  return;
}
// Inisialisasi RTC
if (! rtc.begin()) {
  Serial.println("Couldn't find RTC");
  while (1);
}

if (! rtc.isrunning()) {
  Serial.println("RTC is NOT running!");
  // Baris berikut menyetel RTC ke tanggal & waktu sketsa ini
  disusun
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  // Baris ini menetapkan RTC dengan tanggal & waktu yang jelas,
  misalnya
  // January 21, 2014 pada 3am kamu dapat memanggilnya:
  // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
}
}

void loop () {
  DateTime now = rtc.now();
  sprintf(Time, "%02d:%02d:%02d", now.hour(), now.minute(),
now.second());
  sprintf(Date, "%02d/%02d/%02d", now.day(), now.month(),
now.year());
  Serial.print(Time); // print ke serial monitor
  Serial.print("\t");
  Serial.println(Date);

  myFile = SD.open("Coba11.txt", FILE_WRITE); //membuka file
  dan menulis isi filenya
  if (myFile) {
    myFile.print(Time); // print ke file yang ada di micro SD
    myFile.print(",");
    myFile.println(Date);
    myFile.close(); // tutup file
  }
}

```

```

}
//jika file tidak dapat dibuka, print error.
else {
    Serial.println("error opening test.txt");
}
    delay(1000);
}

```

A.5 Listing Program LCD Keypad

```

#include <LiquidCrystal.h>
#include <phi_interfaces.h>
#include <phi_prompt.h>
byte keypad_type=Analog_keypad;
char mapping[]={ 'R','U','D','L','S' };
byte pins[]={ PIN_KEYPAD };
int values[]={ 5, 105, 260, 415, 640 };
phi_analog_keypads analogKeypad(mapping, pins, values,
BUTTONS_PER_ROW, BUTTONS_PER_COLUMN);
multiple_button_input * keypads[]={ &analogKeypad,0 };
char up_keys[]={ "U" };
char down_keys[]={ "D" };
char left_keys[]={ "L" };
char right_keys[]={ "R" };
char enter_keys[]={ "S" };
char                                     *
function_keys[]={ up_keys,down_keys,left_keys,right_keys,enter_
keys };
LiquidCrystal                               lcd
(LCD_RS,LCD_EN,LCD_D4,LCD_D5,LCD_D6,LCD_D7);
int global_style=109;

void setup() {
    pinMode(pinCS,OUTPUT);
    Serial.begin(9600);
    arduInterfaceSetup();//Setup Keypad-LCD
    pinMode(PIN_LED_LIVE, OUTPUT);
    setupMenuReleComb();
    subMenuInfo();}

```

```

void arduInterfaceSetup(){
    lcd.begin(LCD_COLUMNS, LCD_ROWS);
    init_phi_prompt(&lcd,keypads,function_keys,
LCD_COLUMNS, LCD_ROWS, '~'); }
//MenuReleComb
const char ReleCombMn0[] PROGMEM="Trip-Reset";/
const char ReleCombMn1[] PROGMEM="Display";
const char ReleCombMn2[] PROGMEM="Status";
const char ReleCombMn3[] PROGMEM="Info";
const char* const ReleCombMnItems[] PROGMEM =
{ReleCombMn0,      ReleCombMn1,      ReleCombMn2,
ReleCombMn3};
phi_prompt_struct mainMenu;

void mainMenuRele(){
    switch (menuId.mainMenu)
    {
        case MAIN_MENU_IDX:
            menuReleComb();
            break;
        case TRIP_RESET_MENU_IDX:
            subMenuTripReset("<<-Trip-Reset->>");
            break;
        case DISPLAY_MENU_IDX:
            subMenuDisplay();
            break;
        case STATUS_MENU_IDX:
            subMenuStatus();
            break;
        case INFO_MENU_IDX:
            subMenuInfo();
            break;
        default:
            break;
    }
}

void setupMenuReleComb(){
    mainMenu.ptr.list=(char*)&ReleCombMnItems;

```

```

    mainMenu.low.i=0;
    mainMenu.high.i=2;
    mainMenu.width=LCD_COLUMNS-
((global_style&phi_prompt_arrow_dot)!=0)-
((global_style&phi_prompt_scroll_bar)!=0);
    mainMenu.step.c_arr[0]=LCD_ROWS-1;
    mainMenu.step.c_arr[1]=1;
    mainMenu.step.c_arr[2]=0
    mainMenu.step.c_arr[3]=LCD_COLUMNS-4-
((global_style&phi_prompt_index_list)!=0);    mainMenu.col=0; //
Display menu at column 0
    mainMenu.row=1; // Display menu at row 1
    mainMenu.option=global_style; // Option 0, display classic list,
option 1, display 2X2 list, option 2, display list with index, option
3, display list with index2.
}

```

```

void menuReleComb()
{
    int menu_pointer_1=0; // This stores the menu choice the user
made.
    lcd.clear(); // Refresh menu if a button has been pushed
    center_text("Main Menu");//Menu Title

```

```

    select_list(&mainMenu); // Use the select_list to ask the user to
select an item of the list, that is a menu item from your menu.

```

```

    menu_pointer_1=mainMenu.low.i; // Get the selected item number
and store it in the menu pointer.

```

```

    switch (menu_pointer_1) // See which menu item is selected and
execute that correS_Pond function

```

```

    {
        case 0:
            menuId.mainMenu = TRIP_RESET_MENU_IDX;
            menuId.subMenu = 1;
            break;
        case 1:
            menuId.mainMenu = DISPLAY_MENU_IDX;
            menuId.subMenu = 1;
            break;

```

```

    case 2:
        menuId.mainMenu = STATUS_MENU_IDX;
        menuId.subMenu = 1;
        break;
    case 3:
        menuId.mainMenu = INFO_MENU_IDX;
        menuId.subMenu = 1;
        break;
    default:
        break;
}
}

void subMenuDisplay(){
    byte currentKey;
    String Sval;
    lcd.clear(); // Refresh menu if a button has been pushed

    //Tampilkan pada baris ke 1
    Sval = String("Set:");
    Sval = String(Sval + RelayShortCirt.getSetting());
    Sval = String(Sval + "/");
    Sval = String(Sval + RelayOverLoad.getSetting()); //EmonLib
    Arus
    Sval = String(Sval + " A");
    lcd.setCursor(0,0); //posisikan kursor pada baris 1 kolom 1
    lcd.print(Sval);

    //Tampilkan Nilai Tegangan pada baris ke 2
    Sval = String("V/I:");
    Sval = String(Sval + monitoringArusDanTegangan.Vrms);
    //EmonLib Tegangan 09/05/2017
    Sval = String(Sval + "/");
    Sval = String(Sval + RelayShortCirt.getValue()); //EmonLib Arus
    lcd.setCursor(0,1); //posisikan kursor pada baris 2 kolom 1
    lcd.print(Sval);

    currentKey = analogKeypad.getKey(); // Use phi_keypads object
    to access the keypad

```

```
switch (currentKey) // See which menu item is selected and execute
that correS_Pond function
```

```
{
    case 'S':
        menuId.mainMenu = MAIN_MENU_IDX;
        menuId.subMenu = 1;
        return;
    break;
    default:
        break;
}
}
```

```
void subMenuStatus(){
    // menu untuk menampilkan tgl dan waktu pada baris pertama
    // status relay pada baris kedua
    byte currentKey;
    String Sval;
    lcd.clear(); // Refresh menu if a button has been pushed
```

```
//Tampilkan pada baris ke 1
//DateTime SaatIni = rtc.now();
Sval = String(SaatIni.day());//ambil tanggal
Sval = String(Sval + "/");
Sval = String(Sval + SaatIni.month());//ambil bulan
Sval = String(Sval + "/");
Sval = String(Sval + (SaatIni.year()-2000));///ambil tahun
Sval = String(Sval + " ");
Sval = String(Sval + SaatIni.hour());///ambil jam
Sval = String(Sval + ":");
Sval = String(Sval + SaatIni.minute());///ambil menit
Sval = String(Sval + ":");
Sval = String(Sval + SaatIni.second());///ambil detik
```

```
lcd.setCursor(0,0);
lcd.print(Sval);
```

```
//Tampilkan status relay pada baris ke 2
lcd.setCursor(0,1);
```

```

lcd.print(statusRelay());
currentKey = analogKeypad.getKey(); // Use phi_keypads object
to access the keypad
switch (currentKey) // See which menu item is selected and execute
that correS_Pond function
{
    case 'S':
        menuId.mainMenu = MAIN_MENU_IDX;
        menuId.subMenu = 1;
        return;
    break;
    default:
        break;
}
}
}

```

```

String statusRelay(){
    byte _state;
    String Sval;
    _state = RelayShortCirt.getState();
    if (_state < RelayOverLoad.getState())_state =
RelayOverLoad.getState(); //ambil nilai status tertinggi 25/05/2017
    Sval = String("STS:");
    if (_state == STATUS_OK)return (Sval = String(Sval + "OK"));
    else if (_state >= STATUS_TRIP){
        Sval = String(Sval + "TRP->");
        _state = _state - STATUS_TRIP;
    }
    if (_state > STATUS_OK){//Trip atau belum trip dengan beberapa
status
        //check status Relay
        switch (_state){
            case STATUS_OVER:
                return String(Sval + "OVR");
            break;
            case STATUS_UNDER:
                return String(Sval + "UDR");
            break;
            case EQL_ACTION:

```



```

        return String(Sval + "EQL");
    break;
    case STATUS_TEST_LOCAL:
        return String(Sval + "LOCAL");
    break;
    case STATUS_TEST_KEY:
        return String(Sval + "KEYPAD");
    break;
    case STATUS_TEST_REMOTE:
        return String(Sval + "REMOTE");
    break;
    default:
        break;
    }
}
}

void subMenuInfo(){
char infoMsg[]="by: ..... ";
char buffer[15];
lcd.clear();
lcd.noBlink();
center_text("Protection Relay"); // display judul
for (byte i=0;i<strlen(infoMsg);i++)
{
    scroll_text(infoMsg,buffer,14,i-14);
    lcd.setCursor(1,1);
    lcd.print(buffer);
    wait_on_escape(300);
}
menuId.mainMenu = DISPLAY_MENU_IDX;
menuId.subMenu = 1;
}

void subMenuTripReset(char* judulMenu)
{
    byte currentKey;
    lcd.clear(); // Refresh menu if a button has been pushed
    center_text(judulMenu);//Menu Title

```

```

    currentKey = analogKeypad.getKey(); // Use phi_keypads object
    to access the keypad
    switch (currentKey) // See which menu item is selected and execute
    that correS_Pond function
    {
        case 'U':
            //menuId.mainMenu = TRIP_RESET_MENU_IDX;
            menuId.subMenu--;
            if (menuId.subMenu < TRIP_MENU_IDX) menuId.subMenu =
            BACK_MENU_IDX;//roll over to Back_maneu
            break;
        case 'D':
            menuId.subMenu++;
            if (menuId.subMenu > BACK_MENU_IDX) menuId.subMenu
            = TRIP_MENU_IDX;//roll over to Trip_menu
            break;
        case 'S':
            if (menuId.subMenu == TRIP_MENU_IDX){
                menuId.mainMenu = 0;
                RelayShortCirt.setState(STATUS_TEST_KEY);//Test trip
                dari keypad
                RelayOverLoad.setState(STATUS_TEST_KEY);//Test trip
                dari keypad, 25/05/2017
                return;//ini untuk test trip
            }
            if (menuId.subMenu == RESET_MENU_IDX){
                menuId.mainMenu = 0;
                RelayShortCirt.setReset(true);//Reset dari keypad
                RelayOverLoad.setReset(true);//Reset dari keypad, 25/05/2017
                return;//ini untuk Reset
            }
            if (menuId.subMenu == BACK_MENU_IDX){
                menuId.mainMenu = 0;
                return;
            }
            break;
        default:
            break;
    }
}

```


-----Halaman ini sengaja dikosongkan-----

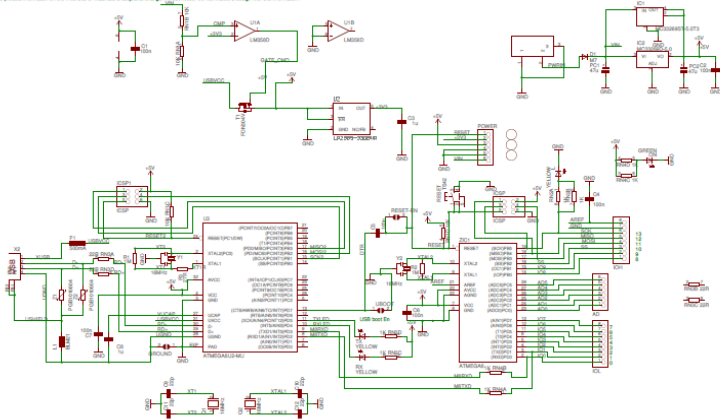
LAMPIRAN B

B.1 *DATASHEET* ARDUINO UNO

Arduino™ UNO Reference Design

Aradigm Designs ARE PROVIDED "AS-IS" AND "WITH ALL FAULTS". ANY IMPLIED WARRANTIES, INCLUDING ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE HEREBY DISCLAIMED.

Aradigm may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Aradigm reserves the right to change specifications and product descriptions at any time without notice. Do not finalize a design with this information.



Technical Specification



EAGLE files: [arduino-damianovs-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C:** 4 (SDA) and 5 (SCL). Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

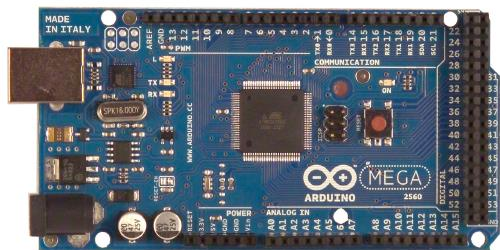
- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

B.2 DATASHEET ARDUINO MEGA 2560



Arduino Mega 2560 Datasheet



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH

value, the LED is on, when the pin is LOW, it's off.

- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I²C (TWI) and SPI communication. The Arduino software includes a `Wire` library to simplify use of the I²C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It

communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

B.3 DATASHEET RS485

RS-485 module for Arduino (MAX485)

Click photo above for details, then hover over upper right for more photos.

This module interfaces an Arduino or similar microcomputer to RS-485. RS485 is used for Serial Communications over longer distances than direct RS232 or TTL, and supports multiple units on the same bus (Multi-Drop).

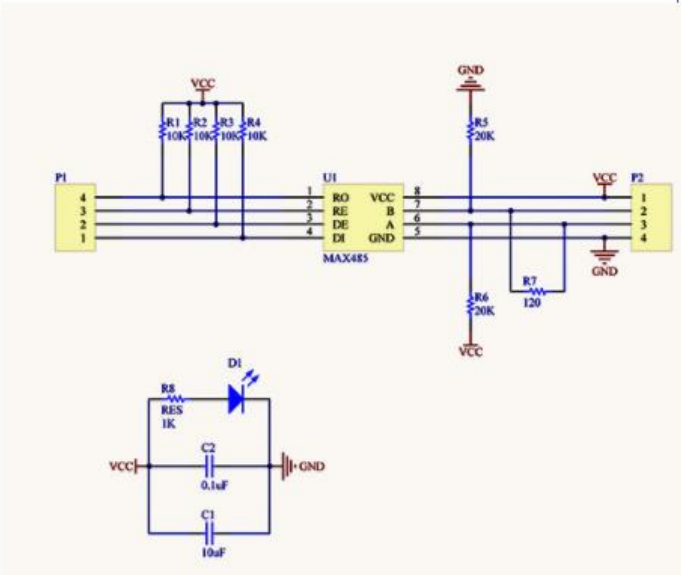
- Multiple Units can be connected to the same RS-485 wiring.
- All Chip pins are brought out for proper controls
- Working voltage: 5V
- Board size: 44 (mm) x14 (mm)

THIS USB INTERFACE for use with PC's is available to connect by RS-485 (click)

How-To Information Link HERE:

See Example RS485 Network Diagram below.

SCHEMATIC OF THIS MODULE:



Selection Table

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/DRIVER ENABLE	QUIESCENT CURRENT (µA)	NUMBER OF RECEIVERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8
MAX487	Half	0.25	Yes	Yes	Yes	120	128	8
MAX488	Full	0.25	Yes	No	No	120	32	8
MAX489	Full	0.25	Yes	No	Yes	120	32	14
MAX490	Full	2.5	No	No	No	300	32	8
MAX491	Full	2.5	No	No	Yes	300	32	14
MAX1487	Half	2.5	No	No	Yes	230	128	8

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

19-0122; Rev 10; 9/14

B.4 DATASHEET DS1307

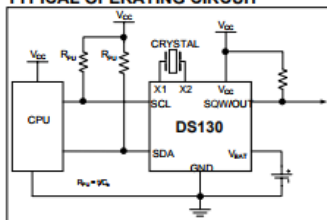


DS1307 64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

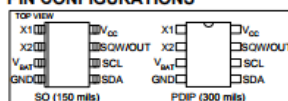
TYPICAL OPERATING CIRCUIT



FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307ZT&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZNT&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/RoHS-compliant package.

*A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial	0°C to +70°C
Industrial	-40°C to +85°C
Storage Temperature Range	-55°C to +125°C
Soldering Temperature (DIP, leads)	+260°C for 10 seconds
Soldering Temperature (surface mount)	Refer to the JPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V _{CC}		4.5	5.0	5.5	V
Logic 1 Input	V _{HI}		2.2		V _{CC} + 0.3	V
Logic 0 Input	V _{LI}		-0.3		+0.8	V
V _{BAT} Battery Voltage	V _{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I _{LI}		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I _{LO}		-1		1	μA
Logic 0 Output (I _{OL} = 5mA)	V _{OL}				0.4	V
Active Supply Current (f _{SCL} = 100kHz)	I _{CCA}				1.5	mA
Standby Current	I _{CCS}	(Note 3)			200	μA
V _{BAT} Leakage Current	I _{BATLKG}			5	50	nA
Power-Fail Voltage (V _{BAT} = 3.0V)	V _{PF}		1.216 x V _{BAT}	1.25 x V _{BAT}	1.284 x V _{BAT}	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V _{BAT} Current (OSC ON); SQW/OUT OFF	I _{BAT1}			300	500	nA
V _{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I _{BAT2}			480	800	nA
V _{BAT} Data-Retention Current (Oscillator Off)	I _{BATDR}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}		0	100		kHz
Bus Free Time Between a STOP and START Condition	t _{BUF}		4.7			μs
Hold Time (Repeated) START Condition	t _{HOLD:STA}	(Note 4)	4.0			μs
LOW Period of SCL Clock	t _{LOW}		4.7			μs
HIGH Period of SCL Clock	t _{HIGH}		4.0			μs
Setup Time for a Repeated START Condition	t _{SUSTA}		4.7			μs
Data Hold Time	t _{HOLD:DAT}		0			μs
Data Setup Time	t _{SUDAT}	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t _r				1000	ns
Fall Time of Both SDA and SCL Signals	t _f				300	ns
Setup Time for STOP Condition	t _{SUSTO}		4.7			μs

CAPACITANCE(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C _{IO}				10	pF
Capacitance Load for Each Bus Line	C _B	(Note 7)			400	pF

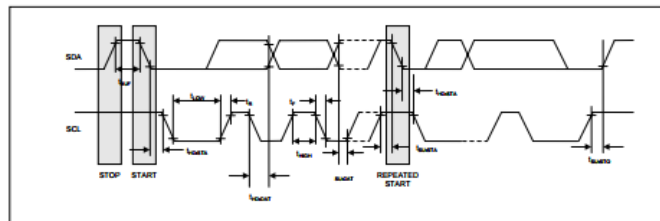
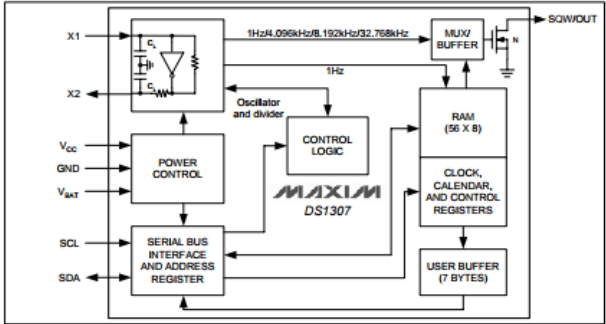
Note 1: All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** I_{CCS} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 5:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{DD(SDA)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 6:** The maximum t_{SUDAT} only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.**Note 7:** C_B—total capacitance of one bus line in pF.**TIMING DIAGRAM**

Figure 1. Block Diagram



B.5 DATASHEET SCT 13-010



Product Specification

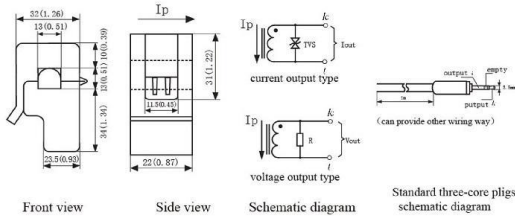
Date:2015-8-7

Product Name	Current transformer	Model	SCT013-010
Characteristics:Opening size 13mm*13mm,1m leading wire, standard Φ3.5 three-core plug output. Have two kinds of output type: Current output type and voltage output type.			
Purpose: Used for current measurement, monitor and protection for AC motor,lighting equipment, air compressor etc			
Technical Data			
I_{PN}	Rated input	0-10A	
I_{PM}	Max. detection input		
I_{OUT}	Rated output	0-1V	
X	Accuracy	±1%	
E_L	Linearity	≤0.2%	
N	Turns ratio	1:1800	
Φ	Phase shift		
R_L	Max.Sampling resistance		
V_{FN}	Work voltage	660V	
f	Work frequency	50-1KHz	
T_A	Operating temperature	-25..+70℃	
T_S	Storage temperature	-40..+85℃	
Vd	Dielectric strength, 50 Hz, 1 min	3KV	

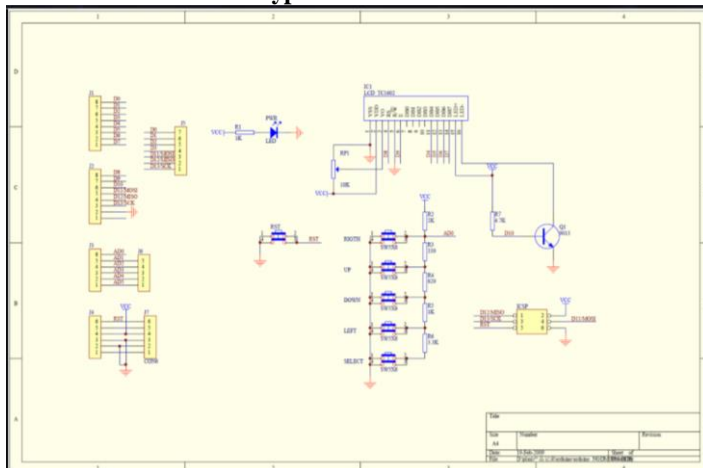


Fire resistance	UL94-V0
Material of core	Ferrite
Mounting type	Suspension
Weight	55g

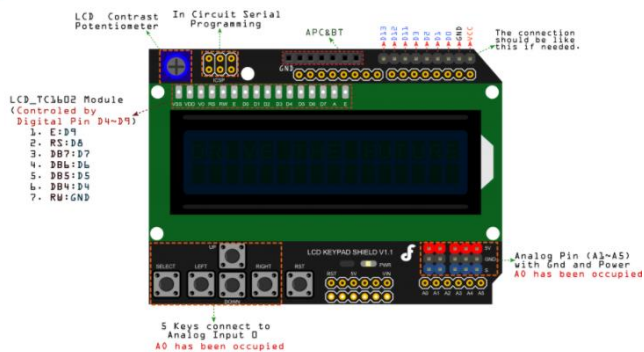
Dimension (mm(in). 1 mm= 0.0394 inch)



B.6 DATASHEET LCD Keypad Shield



Pinout



Instruction for D4 To D10 and Analog Pin 0		
Pin	Function	Instruction
Digital 4(D4)		
Digital 5(D5)		
Digital 6(D6)		
Digital 7(D7)		
Digital 8(D8)		
Digital 9(D9)		
Digital 10(D10)		
Analog 0(A0)		

B.7 Datasheet UPS

SPESIFIKASI			
MODEL		IF-650 WA	IF-1200 WA
CAPACITY	VA	650VA	1200VA
INPUT	Voltage	110VAC/120VAC or 220VAC / 230VAC /240VAC	
	Voltage Range	81-145VAC or 162-290VAC	
OUTPUT	Voltage Regulation (Batt. Mode)	+/- 10%	
	Frequency	50Hz or 60Hz	
	Frequency Regulation (Batt. Mode)	+/- 1Hz	
	Output Waveform	Simulated Sine Wave	
BATTERY	Battery Type	12 V/8.2AH x 1	12 V/8.2AH x 2
	Recharge Time	6-8 hours to 90% after complete discharge	
TRANSFER TIME	Typical	2-6 ms	
INDICATOR (*Note 1)	AC Mode	Green LED lighting	
	Battery Mode	Yellow LED Flashing	
	Fault Mode	Red LED Lighting	
AUDIBLE ALARM	Backup Mode	Sounding every 10 seconds	
	Low Battery	Sounding every 1 second	
	Overload	Sounding every 0.5 second	
	Fault	Continuously sounding	
PROTECTION	Full Protection	Discharge, overcharge, and overload protection	
PHYSICAL	Dimension (mm), LXWXH	298x101x142	353x149.3x162
	Operating Environment	0-90% RH @ 0- 40°C (non-condensing)	
ENVIRONMENT	Noise Level	Less than 40dB	

Catatan 1: untuk model LCD silahkan lihat pilihan "3. LCD pada halaman 3

LAMPIRAN C

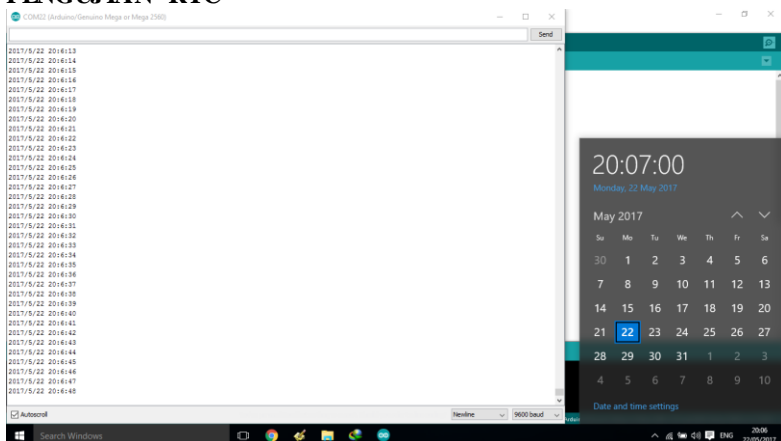
C.1 ALAT ADAPTIF RELAI



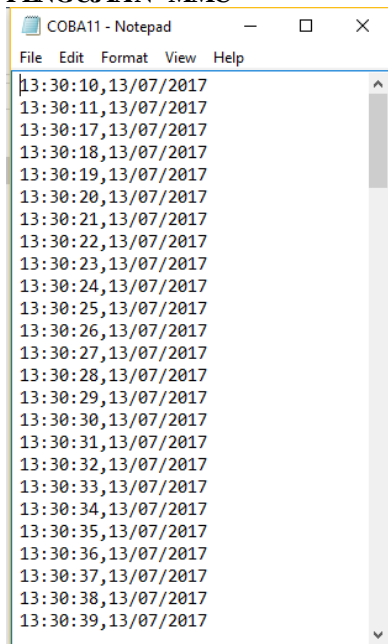
C.2 PENGUJIAN SENSOR ARUS DAN TEGANGAN



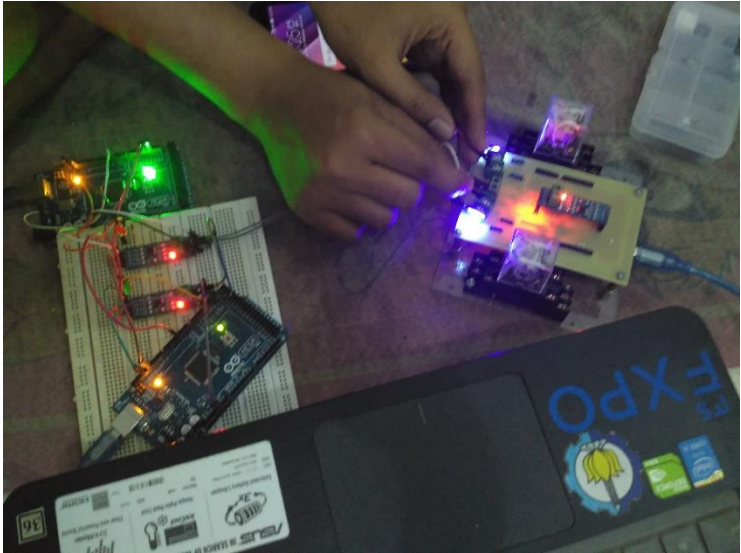
C.3 PENGUJIAN RTC



C.4 PENGUJIAN MMC



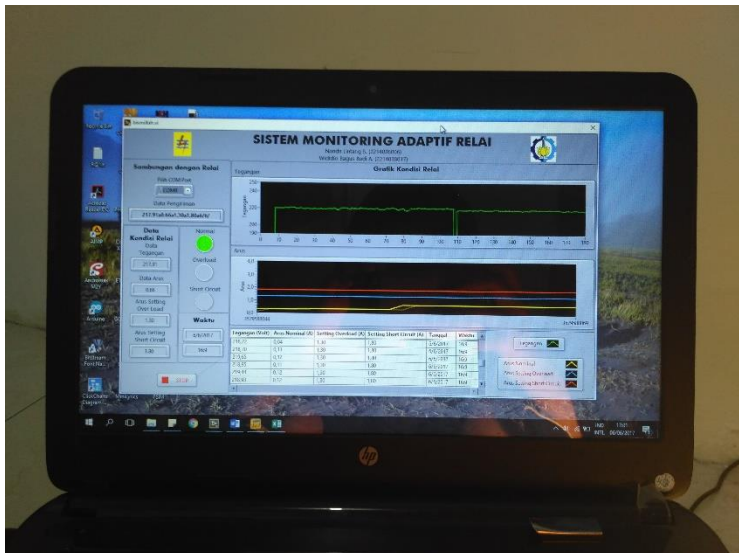
C.5 PENGUJIAN RELAY INDIKATOR



C.6 PENGUJIAN RELAY UTAMA



C.7 PENGUJIAN LABVIEW



DAFTAR RIWAYAT HIDUP



Nama : Wididio Bagus Budi Arto
TTL : Tuban, 14 Desember 1995
Jenis Kelamin : Laki - laki
Agama : Islam
Alamat : JL. Gayungan Pasar no. 27
Surabaya
Telp/HP : 083831436277
E-mail : wididiobagus@yahoo.co.id

RIWAYAT PENDIDIKAN

1. 2002 – 2008 : SDN Gayungan 1 Surabaya
2. 2008 – 2011 : SMP Negeri 22 Surabaya
3. 2011 – 2014 : SMA Negeri 16 Surabaya
4. 2014 – 2017 : D3 Teknik Elektro Otomasi, Program Studi Teknik Listrik – Fakultas Vokasi Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT PLN (Persero) APD Jawa Timur Area Surabaya Selatan

PENGALAMAN ORGANISASI

1. Staff Departemen Big Event HIMAD3TEKTRO 2015-2016
2. Kepala Departemen Big Event HIMAD3TEKTRO 2016-2017

-----Halaman ini sengaja dikosongkan-----